PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10

A2

(11) International Publication Number: **WO 98/52158**

(43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01388

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/046,543	15 May 1997 (15.05.97)	US
09/078,031	13 May 1998 (13.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47-53 Cannon Street, London EC4M 5SQ (GB).

(72) Inventors: EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berks. RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephens; The Elms, School Road, Broughton, Cambs. PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts. WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham GU20 6DL (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

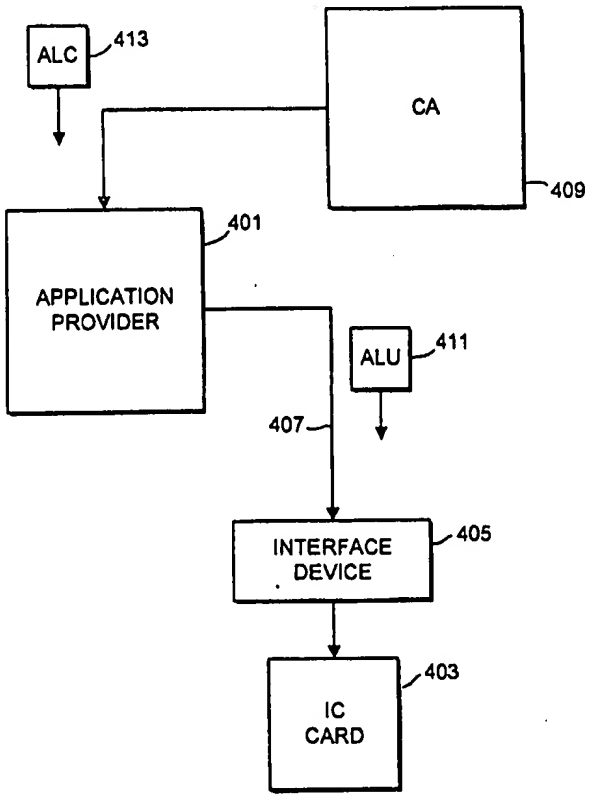
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: INTEGRATED CIRCUIT CARD WITH APPLICATION HISTORY LIST

(57) Abstract

There is provided an integrated circuit card for loading an application copy thereon and a method of loading an application copy onto the integrated circuit card, wherein the application copy is one of a plurality of copies of an application. The application copy has an associated application identifier that uniquely identifies the application from other applications and an application copy number that is unique for each copy of the application. The integrated circuit card includes a microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto the integrated circuit card. The method includes receiving by the integrated circuit card the application copy, the application identifier, and the application copy number; determining by the integrated circuit card whether the application identifier and the application copy number are contained in the application history list area; and failing to load the application copy by the integrated circuit card if the application identifier and the application copy number are contained in the application history list area.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

INTEGRATED CIRCUIT CARD WITH APPLICATION HISTORY LIST

BACKGROUND OF INVENTION

Integrated circuit (IC) cards are becoming increasingly used for many different purposes in the world today, principally because they are ideal tools for the delivery of distributed, secure information processing at a low cost. An IC card, also called a "smart card," is a card typically the size of a conventional credit card, but which contains a computer chip on the card. The computer chip on the IC card typically includes a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an input/output (I/O) mechanism, and other circuitry to support the microprocessor in its operations. The computer chip can execute one or more applications stored on the card. Examples of applications that IC cards are being used to store and execute include credit/debit, electronic money/purse, telephone calling card, and loyalty reward applications.

When an application is initially loaded onto an IC card, the application may include data that is associated with the application. Such data may include, for example, data that identifies the cardholder, such as the cardholder's name and account number. Additionally, the associated data may also include a promotional or bonus value provided by the application provider to the cardholder for loading the application. For example, with a telephone calling card application, an application provider may provide a certain amount of free calling time. As another example, with an electronic purse application, an application provider may provide bonus electronic cash. As yet another example, with a frequent flyer loyalty application, an application provider may provide free miles.

The use of application data to provide promotional or bonus value creates a potential problem for the IC card manufacturer and the application provider regarding the integrity of loading applications. A solution is needed to prevent a cardholder from intentionally or unintentionally copying an application when it is first loaded, and reloading the application thereafter to reload the value in the data associated with the application. By repeated reloading of an application, a cardholder may potentially obtain an unlimited amount of promotional or bonus value to which he or she is not entitled. At the same time, however, cardholders may be required to reload an application for legitimate reasons, such as for updating an application.

Accordingly, a need exists for a method of loading an application onto an IC card such that a cardholder is prevented from illegitimately reloading an application once it has been loaded onto the IC card.

The foregoing technical challenges and needs are addressed by embodiments in accordance with the invention which provides technical solutions.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment of the present invention, there is provided a method of loading an application copy onto an integrated circuit card, wherein the application copy is one of a plurality of copies of an application. The application copy has an associated application identifier that uniquely identifies the application from other applications and an application copy number that is unique for each copy of the application. The integrated circuit card includes a

microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto the integrated circuit card. The method includes receiving by the integrated circuit card the application copy, the application identifier, and the application copy number; determining by the integrated circuit card whether the application identifier and the application copy number are contained in the application history list area; and failing to load the application copy by the integrated circuit card if the application identifier and the application copy number are contained in the application history list area.

As it is used in this specification and the appended claims, the term "unique" to refer to application copy numbers refers to two types of numbers: (1) non-random numbers that are actually determined to be unique, and (2) random numbers that are determined to be probabilistically unique for a given cardholder.

The method in accordance with the preferred embodiment of the present invention may further include the steps of allocating a predetermined portion of the memory for the application history list area; determining by the integrated circuit card whether the application history list area is full; and failing to load the application copy if the application history list is full.

The method in accordance with the preferred embodiment of the present invention may further include the step of adding the application identifier and the application copy number to the application history list area if the application identifier and the application copy number are not contained in the

application history list area. Thus, once a copy of an application is loaded onto the integrated circuit card, the application identifier and the application copy number associated with the copy of the application are stored in the application history list area for future checking.

The method in accordance with the preferred embodiment of the present invention may also provide a mechanism by which application providers not concerned with repeated loading of applications may circumvent storage of the application identifier and the application copy number in the application history list area. For example, an application copy number of zero can be used to signify that an application may be reloaded as often as desired. Accordingly, the method of the preferred embodiment of the present invention may further include the step of adding the application identifier and the application copy number to the application history list area if the application identifier and the application copy number are not contained in the application history list area and the application copy number is not zero.

The application copy may include both application code and application data. The application identifier and the application copy number may be contained in the application data.

Preferably, the application copy, the application identifier, and the application copy number are transmitted to the integrated circuit card by an application provider. Preferably, before transmitting the application copy to the integrated circuit card, the application provider encrypts at least a portion of the application copy. It is also preferred that an application provider transmit a key

transformation unit, which includes information relating to the encryption of the encrypted portion of the application copy. It is further preferred that the integrated circuit card has a first public key pair and that the application provider encrypts the key transformation unit with the public key of the first public key pair before transmitting the key transformation unit to the integrated circuit card.

When the application provider encrypts the key transformation unit with the public key of the first public key pair, the integrated circuit card may decrypt the encrypted key transformation unit with the secret key of the first public key pair. Once the key transformation unit is decrypted, the integrated circuit card may decrypt the application copy using the information contained in the decrypted key transformation unit.

It is also preferred that the application provider has a second public key pair and that the application provider form a signed application copy by encrypting the application copy with the secret key of the second public key pair. The application provider may then transmit both the application copy and the signed application copy to the integrated circuit card.

It is further preferred that the application provider register the public key of the second public key pair with a certification authority, which has a third public key pair. The certification authority may then provide a certificate to the application provider by encrypting the public key of the second public key pair with the secret key of the third public key pair. The application provider may transmit the certificate to the integrated circuit card.

When a certificate is transmitted to the integrated circuit card, the

integrated circuit card may obtain the public key of the second key pair by decrypting the certificate using the public key of the third public key pair. The integrated circuit card may then verify the signed application copy using the public key of the second public key pair. The integrated circuit card may fail to load the application copy if the signed application copy is not verified.

In accordance with another preferred embodiment of the present invention, there is provided an integrated circuit card that includes a microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers, each application identifier and each application copy number being associated with an application copy. The application copy is one of a plurality of copies of an application. Each application identifier uniquely identifies an application from other applications, and each application copy number uniquely identifies an application copy from other application copies. The integrated circuit card of the invention further includes means for determining whether an application identifier and an application copy number associated with an application copy to be loaded into the memory area are contained in the application history list area and means for failing to load the application copy to be loaded if the associated application identifier and the associated application copy number are contained in the application history list area.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments in accordance with the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1 is a schematic representation of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 2 is a perspective view of an IC card and terminal in accordance with a preferred embodiment of the present invention;

Fig. 3 is a functional block diagram of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 4 is a diagram of a system for remotely loading an application from an application provider onto an IC card in accordance with a preferred embodiment of the present invention;

Fig. 5 is a schematic representation of an application load unit in accordance with a preferred embodiment of the present invention;

Fig. 6 is a flowchart of exemplary steps for processing the application load unit of Fig. 5 in accordance with a preferred embodiment of the present invention; and

Fig. 7 is a flowchart illustrating exemplary steps of a file loading routine, which may be implemented by the operating system of an IC card in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 provides a schematic representation of a typical IC card 10 that can be used with the presently claimed invention. The IC card 10 includes an integrated circuit 12 having one or more electrical contacts 14 connected to the integrated circuit 12.

Fig. 2 shows an example of a device with which the IC card 10 communicates. As used in this specification and the appended claims, the terms "interface device" and "terminal" shall be used to generically describe devices with which an IC card may communicate. A typical terminal 20, as shown in Fig. 2, includes a card reader 22, a keypad 24, and a display 26. The keypad 24 and the display 26 allow a user of the IC card 10 to interact with the terminal. The keypad 24 allows the user to select a transaction, to enter a personal identification number ("PIN"), and to enter transactional information. The display 26 allows the user to receive informational messages and prompts for data entry. Other types of terminals may include IC card-compatible ATM machines and telephones.

Fig. 3 provides a functional block diagram of the integrated circuit 12. At a minimum, the integrated circuit 12 includes a processing unit 100 and a memory unit 110. Preferably, the integrated circuit 12 also includes control logic 150, a timer 160, security circuitry 170, input/output ports 180, and a co-processor 190. The control logic 150 provides, in conjunction with the processing unit 100, the control necessary to handle communications between the memory unit 110 and input/output ports 180. The timer 160 provides a timing reference signal for the processing unit 100 and the control logic 150. The security circuitry 170 preferably

provides fusible links that connect the input/output ports 180 to internal circuitry for testing during manufacturing. The fusible links are burned after completion of testing to limit later access to sensitive circuit areas. The co-processor 190 provides the ability to perform complex computations in real time, such as those required by cryptographic algorithms.

The memory unit 110 may include different types of memory, such as volatile and non-volatile memory and read-only and programmable memory. For example, as shown in Fig. 3, the memory unit 110 may include read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), and random-access memory (RAM).

The memory unit 110 stores IC card data such as secret cryptographic keys and a user PIN. The secret cryptographic keys may be any type of well-known cryptographic keys, such as the private keys of public-key pairs. Preferably, the secret cryptographic keys are stored in a secure area of ROM or EEPROM that is either not accessible or has very limited accessibility from outside the IC card.

The memory unit 110 also stores the operating system of the IC card. The operating system loads and executes IC card applications and provides file management and other basic card services to the IC card applications. Preferably, the operating system is stored in ROM.

In addition to the basic services provided by the operating system, the memory unit 110 may also include one or more IC card applications. For example, if the IC card is to be used as an electronic cash card, an application

called MONDEX™ PURSE (from Mondex International Limited) might be included on the IC card, which loads an electronic value of a certain currency from a user's account in a financial institution onto the IC card. Preferably, the operating system of the IC card 10 should support multiple applications, such as the MULTOS™ operating system from Mondex International Limited.

An IC card application may include both program and associated data files, which are typically stored in EEPROM. The application program may be written either in the native programming code of the processing unit 100 or it may be written in a higher level language that must be translated before it is executed on the processing unit 100. An example of such a higher level language for use on IC cards is the MULTOS™ Executable Language (MEL). Advantageously, by using a higher level language such as MEL, an application program is capable of running on multiple hardware platforms without any need for re-writing.

Because IC cards typically have limited memory capacity due to the size and cost restraints of placing memory on the IC cards, an IC card may also have primitives stored in ROM, which are subroutines that perform frequently used functions or procedures, such as mathematical functions. The primitives are usually written in the native language of the processing unit 100 so that they can be executed very quickly.

In Fig. 4, there is shown a diagram of a system for remotely loading an application from an application provider 401 onto an IC card 403. The application provider 401 may be a card issuer, a bank, or any other entity that provides application loading services. The IC card 403 communicates with the

application provider 401 through an interface device 405, which may be a bank terminal, an ATM, or any other device that communicates with an IC card. The application provider 401 and the interface device 405 communicate by way of a data conduit 407, which can be a telephone line, a cable line, a satellite link, an Internet connection, an intra-net connection, or any other type of communications link.

When loading applications onto an IC card remotely, an application provider is required to address several security issues. First, an application provider must ensure that an application is sent only to the cardholder who is intended to receive the application. Second, the application provider must ensure the privacy of any confidential or trade secret information contained in the applications to be loaded. Third, because the data conduit 407 may be an open link and subject to third parties possibly intercepting or replacing applications being transmitted, an application provider must take security measures to enable the IC card to authenticate the application.

The solutions to these security issues typically involve encryption using symmetric and/or asymmetric cryptography techniques. Symmetric cryptography involves encoding and decoding data using the same mathematical number, called a "key," which must be kept secret. On the other hand, asymmetric cryptography, or "public key" cryptography as it is also called, involves encoding data with one key and decoding data with another key. The two keys are referred to as a key pair, and one of the key pair must be kept secret while the other of the key pair may be publicly distributed. Each key of a key pair may be used to

encode data; however, once data is encoded by using one key, it can only be decoded by using the other key.

In the system of Fig. 4, it is assumed that the application provider 401 and the IC card 403 each have cryptographic key pairs. The generation of cryptographic keys is performed by any manner known by those skilled in the art. The system also utilizes a Certification Authority (CA) 409, which also has a cryptographic key pair. The CA 409 may be any entity that is trusted to keep the secret key of its public key pair private and to authenticate the identity of other entities ___ as, for example, the identity of the application provider 401.

In the system of Fig. 4, the application provider 401 applies for registration of its public key with the CA 409. To do so, the application provider 401 must meet the identification requirements of the CA 409. If the application provider 401 meets these identification requirements, the CA 409 will issue an Application Load Certificate (ALC) 413, which includes the public key of the application provider 401 encoded or "signed" by the secret key of the CA 409. The ALC 413 may be decoded using the public key of the CA 409, which is publicly distributed. Since the CA 409 is trusted to keep its secret key private and to authenticate the identity of the application provider 401, any entity receiving the ALC 413 is assured that the public key contained within the certificate belongs to the application provider 401.

To load an application onto the IC card 403; the application provider 401 transmits an Application Load Unit (ALU) 411 to the interface device 405 via the data conduit 407. The contents of the ALU 411 are shown schematically in

Fig. 5. The ALU preferably includes an Application Unit (AU) 415, a signed Application Unit (AU_s) 417, a Key Transformation Unit (KTU) 419, and the ALC 413.

The AU 415 contains the application code and data that are to be stored on the IC card. Some or all of the application code and data may be encrypted to protect confidential or trade secret portions of the application code and data.

The AU_S 417 is the application code and data AU 415 signed with the secret key of the application provider 401. Using the public key of the application provider 401 provided in the ALC 413, the IC card 403 may decode the AU_S 417 and compare it to the AU 415 to ensure that the AU 415 has not been tampered with during transmission.

The KTU 419 contains information relating to the encrypted portions of the AU 415. This information allows the IC card 403 to decode those encrypted portions so that the application code and data can be accessed by the IC card 403. The KTU 419 is signed with the public key of the IC card 403, which ensures that only the intended IC card 403 can decode the KTU 419 (using the IC card's secret key). Once the KTU 419 is decoded, the IC card 403 may use the information contained in the KTU 419 to decode the encrypted portions of the application code and data of AU 415.

Fig. 6 shows a flow chart of the steps for processing the ALU 411 when it is received by the IC card 403. In step 601, the IC card 403 receives the ALU 411 from the application provider 401. The ALU 411 is placed in the

EEPROM of the IC card 403 along with header information indicating the location in memory of AU 415, AU_s 417, KTU 419 and ALC 413.

In step 603, the ALC 413 is decoded using the public key of the CA 409. The IC card 403 preferably stores in its memory a copy of the CA public key because it may be used in many transactions. Alternatively, the IC card could obtain the public key from a trusted storage location, such as the interface device 405. Once decoded, the ALC 413 provides the IC card 403 with a trusted copy of the public key of the application provider 401.

In step 605, the IC card 403 uses the application provider's public key to verify the AU 415 was not tampered with during transmission. Using the public key of the application provider 401, the IC card 403 decodes the AU_s 417, which was signed with the secret key of the application provider 401. Once the AU_s 417 is decoded, the decoded AU_s 417 is compared to the AU 415. If the two units match, then the AU 415 is verified.

In step 607, the KTU 419, which has been encrypted with the public key of the IC card 403, is decoded using the private key of the IC card 403. In step 609, the information in the decoded KTU 419 is used to decode the encrypted portions of the AU 415. The KTU 419 may contain, for example, either an algorithm or a key for use in decoding the AU 415.

In addition to the security and authentication measures discussed above, other security and authentication measures may also be employed. Additional methods of security and authentication have been addressed, for example, in the related International Patent Application No. PCT/GB98/00531

entitled "Multi-Application IC Card System" by Everett et al., filed February 19, 1998, and US Application entitled "Key Transformation Unit for an IC Card" by Richards et al., filed May 11, 1998. Both of these applications are hereby incorporated by reference to Annex A and Annex B respectively, and Annex C, all attached herewith.

In accordance with a preferred embodiment of the present invention, the data portion of the AU 415 includes an application identifier for the application to be loaded onto the IC card 403 and an application copy number, which is unique for each copy of an application to be loaded onto the IC card 403. As it is used in this specification and the amended claims, the use of the term "unique" in relation to application copy numbers refers both to non-random numbers that are actually determined to be unique and to random numbers that are determined to be probabilistically unique for a given IC card. Preferably, the data portion of the AU 415 containing the application identifier and the application copy number is encoded (and the KTU 419 contains the information necessary to decode this data portion).

Fig. 7 is a flowchart illustrating the steps of a file loading routine that may be implemented by the operating system of the IC card 403 to take advantage of the application identifier and the application copy number contained in the AU 415 to prevent a cardholder from repeatedly loading the same application onto the IC card 403. In the embodiment of Fig. 7, the application copy number is a random number, also called a "random seed." In step 701, the file loading routine receives the file loading command *load_file_command* from the security

manager of the operating system, *OS_Security_Manager*. The *OS_Security_Manager* of the operating system is responsible for verification and decoding of the ALU 411 as discussed with regard to Fig. 6.

In step 703, the application identifier and random seed associated with the application, referred to as *load_file_command.application_id* and *load_file_command.random_seed*, respectively, are checked against entries in an application history list stored on the IC card, referred to as *os_global_data.app_history_list*. The application history list contains entries for each set of application identifier and random seed associated with an application loaded onto the IC card 403. It is preferred that the application history list be stored in a secure area of EEPROM that is not accessible from outside the IC card.

If the application identifier and random seed associated with the application to be loaded are found in the application history list, in step 705, the response status *load_file_response.status* is set to "failed" and the error description *load_file_response.error_cause* is set to "application previously loaded." The error response *load_file_response* is returned to the *OS_Security_Manager*, indicating that the load file routine failed to load the application because the application had previously been loaded onto the IC card.

If the application identifier and random seed associated with the application to be loaded are not found in the application history list, in step 707, the random seed is checked to determine whether it is equal to zero and the application history list is checked to determine whether it is full. A random seed with a value

of zero indicates that the application does not contain any economic value included in its data, and thus may be reloaded as often as desired. If the random seed associated with the application is not zero (indicating there is an economic value included with the application) and the application history list is full, the response status *load_file_response.status* is set to "failed" and the error description *load_file_response.error_cause* is set to "application history list full." In this case, the application cannot be loaded because the application history list is full and, therefore, the application identifier and random seed cannot be added to the application history list for future checking.

If an error condition has not been triggered in steps 703 or 707, in step 711, the directory file record associated with the application is added to the directory file of the IC card -- i.e., the application is loaded onto the IC card 403. In step 713, it is checked whether the random seed is equal to zero. If the random seed is not equal to zero (indicating that there is an economic value included with the application), the application identifier and the random seed are added to the application history list for checking against subsequent applications sought to be loaded onto the IC card. After updating the application history list, the response status *load_file_response.status* is set to "success" and sent to the *OS_Security_Manager*.

If the random seed is equal to zero (indicating that there is no economic value included with the application), the application identifier and random seed are not added to the application history list. Instead, step 717 is skipped, and

the response status *load_file_response.status* is set to "success" and sent to the *OS_Security_Manager*.

Advantageously, the file loading routine of Fig. 7 prevents a cardholder from illegitimately reloading an application. If a cardholder intercepts and copies an application to be loaded onto an IC card, the cardholder cannot later reload the application because, once the application is loaded, the application identifier and random seed are stored permanently on the IC card. If a cardholder attempts to reload the application, the operating system of the IC card will fail to reload the application because the application identifier and random seed of the application will match an entry in the application history list of the IC card.

On the other hand, a cardholder is not prevented from legitimately reloading an application from an application provider. Since an application provider will generate a new random seed for each copy of an application it provides, it will be unlikely for a cardholder to receive a second copy of the application from the application provider with the same random seed. Of course, the application provider must use a random seed of sufficient length to ensure that the probability of any cardholder twice receiving the same random seed is sufficiently unlikely.

Alternatively, instead of using a random number, an application provider may use any unique number associated with copies of applications it provides to each cardholder. For example, an application provider may keep a counter that tracks the number of copies of an application that is has provided. The

application provider may use the value of the counter to provide a unique number each time it provides a copy of the application to a cardholder. The random seed embodiment is preferred, however, because it is easier to manage (i.e., there is no information that is required to be stored or managed).

Although the present invention has been described with reference to certain preferred embodiments, various modifications, alterations, and substitutions will be known or obvious to those skilled in the art without departing from the spirit and scope of the invention, as defined by the appended claims.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A TO THE DESCRIPTION

ANNEX A

MULTI-APPLICATION IC CARD SYSTEM

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application at its personalization stage. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as

both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

The Applicant has recognised therefore, that it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be pre-selected and placed in the memory of the card during is production stage, it would also be beneficial to have the ability to load and delete applications for card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. The Applicant has further recognised that it would be beneficial to have the capability of the IC card system to exchange data among cards, card issuers, system operators and application

providers securely and to load and delete applications securely at any time from either a terminal or remotely over a telephone line, internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity-authentication techniques must be implemented to make sure that applications being sent over the transmission lines are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is highly advantageous since it protects the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. In this regard, to protect against the improper or undesired loading of applications onto IC cards, the Applicant has further recognised that it would be beneficial for the IC card system to have the capability of controlling the loading process and restricting, when necessary or desirable, the use of certain applications to a limited group or number of cards such that the applications are "selectively available" to the IC-cards in the system. This "selective capability" would allow the loading and deleting of applications at, for example, a desired point in time in the card's life cycle. It would also allow the loading of an application only to those cards chosen to receive the selected application.

Accordingly, it is an advantage of a preferred embodiment of the invention that it provides these important features and specifically a secure IC-card system that allows for selective availability of smart card applications which may be loaded onto IC cards.

ANNEX A TO THE DESCRIPTION

These and other advantages are achieved by an embodiment of the present invention which proves an IC card system comprising at least one IC card and an application to be loaded onto the card wherein the IC card contains card personalization date and the application is assigned application permissions data designating which IC card or group of IC cards upon which the application may be loaded. The system checks to determine whether the card's personalization data falls within the permissible set indicated by the application's permissions data. If it does, the application may be loaded onto the card.

In a preferred embodiment, the card personalization data is transferred onto the card by the personalization bureau after the card is manufactured. The data preferably includes data representing the card number, the issuer, product class (i.e., such as gold or platinum cards), and the date on which the card was personalized. The card further preferably contains enablement data indicating whether or not the card has been enabled with personalized data.

In a further preferred embodiment, the IC card secure system checks the enablement data prior to loading an application to determine whether or not the card has been enabled. Preferably, if the card has been enabled, the system checks if the card number, the issuer, the product class and/or the date on which the card was personalized are within the acceptable set indicated by the application's permissions data. If so, the application may be loaded onto the IC card.

ANNEX A TO THE DESCRIPTION

In yet another preferred embodiment, the application's permissions data may contain data representative of a blanket permission such that all cards would pass for application loading.

Further aspects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the three stages in the life of a multi-application IC card in a secure system;

Fig. 2 is a block diagram illustrating the steps of the card manufacture process;

Fig. 3 is a flow diagram illustrating the steps involved in enabling each of the IC cards in the secure system;

Fig. 4 is a block diagram of an IC card chip which can be used in accordance with an embodiment of the invention;

Fig. 5 is a block diagram illustrating the data stored on the IC card as indicated in block 307 of Fig. 3;

Fig. 5A is a schematic of the data structures residing in an IC card and representing personalization data;

ANNEX A TO THE DESCRIPTION

Fig. 6 is a flowchart illustrating the steps of loading an application onto an IC card in the secure system;

Fig. 7 is a flow chart illustrating the checking steps as indicated in block 601 of Fig. 6;

Fig. 8 is a flowchart illustrating the steps undertaken in determining if loading of an application may proceed;

Fig. 9 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system; and

Fig. 10 is a system diagram of entities involved with the use of the IC card once it has been personalized.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

ANNEX A TO THE DESCRIPTION

An embodiment of the present invention provides an IC card system and process which allow the flexibility to load and delete selected applications over the lifetime of a multi-application IC card in response to the needs or desires of the card user, card issuers and/or application developers. A card user who has such a card can selectively load and delete applications as desired if allowed by the card issuer in conjunction with the system operator or Certification Authority ("CA") which controls the loading and deleting process by certifying the transfer of information relating to the process.

By allowing applications to be selectively loaded and deleted from the card, a card issuer can extend additional functionality to an individual IC card without having to issue new cards. Moreover, application developers can replace old applications with new enhanced versions, and applications residing on the same card using a common multiple application operating system may interact and exchange data in a safe and secure manner. For example, a frequent flyer loyalty program may automatically credit one frequent flyer mile to a card user's internal account for every dollar spent with an electronic purse such as the Mondex purse or with a credit/debit application. By allowing the ability to selectively load and delete applications, the card user, subject to the requirements of the card issuer, also has the option of changing loyalty programs as desired.

A card issuer or application developer may intend that a particular application be loaded on only one card for a particular card user in a card system. A regional bank may desire to have a proprietary application reside only on the cards which

ANNEX A TO THE DESCRIPTION

the bank issues. Embodiments in accordance with the present invention would allow for this selective loading and specifically allow for the prevention of loading proprietary applications onto unauthorized cards issued by others.

To achieve these desired objectives, embodiments of the present invention give each card a specific indentity by storing "card personalization data" on the card. Morover, each application to be loaded or deleted on one or more cards in the system is assigned "application permissions data" which specify the cards upon which the applications may be loaded.

The type of personalized data can vary depending upon the needs and requirements of the card system. In the preferred embodiment, described in greater detail below, the personalization data include unique card identification designation data, the card issuer, the product class or type (which is defined by the card issuer) and the date of personalization. However, not all of these data elements are required to be used and additional elements could also be included.

The application permissions data associated with an application, also described in greater detail below, can be a single value in an identity field or could include multiple values in the identity field. For example, the application permissions data in the card issuer field could represent both product class A and product class B from a certain Bank X, indicating that the application could be loaded onto cards designated as product classes A and B issued by Bank X (as indicated in the card product ID field of the card's personalization data).

ANNEX A TO THE DESCRIPTION

In addition, a "global value" could be stored in the issuer field (or other field) of the application permissions data indicating that all IC cards in the system regardless of who issued the card would match this permissions field. In this case, for example, a data value of zero stored in the application permissions card-issuer field will match all of the cards' personalization card-issuer fields.

Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail below.

Card Manufacture

Figure 2 shows the steps necessary in manufacturing an IC card in a secure system. Step 201 manufactures the physical IC card by creating the integrated circuit on silicon and placing it on the card. The integrated circuit chip will include RAM, ROM and EEPROM memories. When the card is first manufactured, a global public key of the system operator (in this case called the Certification Authority (CA)) is stored on each card in ROM in step 203. This will allow the card to authenticate that the source of any message to it is from the CA since the public key on the card will be matched to the CA's secret key.

ANNEX A TO THE DESCRIPTION

More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor.

After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with

ANNEX A TO THE DESCRIPTION

an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the

card identification data, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further described in connection with Figure 5.)

ANNEX A TO THE DESCRIPTION

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus preventing illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that

ANNEX A TO THE DESCRIPTION

accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

ANNEX A TO THE DESCRIPTION

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a

particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load

ANNEX A TO THE DESCRIPTION

cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity: these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit

ANNEX A TO THE DESCRIPTION

application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an "open command" initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card

ANNEX A TO THE DESCRIPTION

has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application

loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern

ANNEX A TO THE DESCRIPTION

due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

ANNEX A TO THE DESCRIPTION

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the

ANNEX A TO THE DESCRIPTION

card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card numbers or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

ANNEX A TO THE DESCRIPTION

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the

ANNEX A TO THE DESCRIPTION

system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The

ANNEX A TO THE DESCRIPTION

cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the

ANNEX A TO THE DESCRIPTION

person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

ANNEX A TO THE DESCRIPTION

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized

ANNEX A TO THE DESCRIPTION

card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or

ANNEX A TO THE DESCRIPTION

delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

ANNEX A TO THE DESCRIPTION

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The applicant hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A TO THE DESCRIPTION

CLAIMS:

1. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for determining whether said card is qualified to accept the loading of said application onto said card.

2. The IC card system of claim 1, wherein said IC card contains card personalization data, and said application is assigned application permissions data representing at least one set of IC cards upon which said application may be loaded.

3. The IC card system of claim 2, wherein said determining means compares said card personalization data with said application permissions data.

4. The IC card system of claim 3, wherein whether said application is loaded onto said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is loaded.

5. The IC card system of any of claims 2 to claim 4, wherein said personalization data comprises data representative of a unique card identification designation.

ANNEX A TO THE DESCRIPTION

6. The IC card system of any of claims 2 to claim 5, wherein said personalization data comprises data representative of a card issuer.

7. The IC card system of any of claims 2 to claim 6, wherein said personalization data comprises data representative of a product class.

8. The IC card system of any of claims 2 to claim 7, wherein said personalization data comprises data representative of a date.

9. An IC card system comprising at least one IC card and an application, wherein said IC card contains personalization data representative of that card and said application is assigned a permissions data set representing at least one IC card upon which said application may be loaded, said system further comprising means for determining whether said personalization data falls within said permissions data set.

10. The IC card system of claim 9 wherein said application is loaded onto said IC card in the event said determining means determines that said personalization data falls within said set.

11. The IC card system of claim 9 or claim 10 wherein said personalization data comprises data representing a card identification designation, and an issuer of said card.

ANNEX A TO THE DESCRIPTION

12. The IC card system of any of claims 9 to claim 11 wherein said personalization data comprises data representing a product class and a date.

13. The IC card system of any of claims 9 to 12 wherein said permissions data set includes a plurality of card identification designations.

14. The IC card system of any of claims 9 to 13 wherein said permissions data set includes one or more issuers of IC cards.

15. The IC card system of any of claims 9 to 14 wherein said permissions data set includes one or more product classes.

16. The IC card system of any of claims 9 to 15 wherein said permissions data set includes a plurality range of dates.

17. The IC card system of any of claims 9 to 16 wherein said permissions data set includes all IC cards which attempt to load the application.

18. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for enabling said card to be loaded with said application.

ANNEX A TO THE DESCRIPTION

19. The IC card system of claim 18 wherein said enabling means comprises means for storing personalization data onto said card.

20. The IC card system of claim 18 wherein said enabling means comprises means for setting an enablement bit.

21. The IC card system of claim 19 wherein said enabling means comprises means for setting an enablement bit.

22. The IC card system of claim 20 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

23. The IC card system of claim 21 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

24. A process for loading an application onto an IC card comprising the step of determining whether said IC card is qualified to accept the loading of said application onto said card.

25. The process of claim 24 wherein said determining step includes the steps of: providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards upon which said application may be loaded;

comparing said personalization data with said permissions data; and

loading said application onto said IC card provided said personalization data falls within said set of cards upon which said application may be loaded.

26. The process of claim 25, wherein said personalization data comprises data representative of a card identification designation.

27. The process of claim 25 or claim 26, wherein said personalization data comprises data representative of a card issuer.

28. The process of any of claims 25 to claim 27, wherein said personalization data comprises data representative of a product class.

29. The process of any of claims 25 to claim 28, wherein said personalization data comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

30. The process of any of claims 25 to claim 29 further comprising the first step of enabling said card to be loaded with said application.

31. The process of claim 30 wherein said enabling step includes the step of storing personalization data onto said card.

32. The process of claim 30 wherein said enabling step includes the step of setting an enablement bit indicating that the card has been enabled.

33. The process of claim 31 wherein said enabling step further includes the step of setting an enablement bit indicating that the card has been enabled.

34. The process of claim 32 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

35. The process of claim 33 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

36. A process for deleting an application from an IC card comprising the step of determining whether said IC card is qualified to delete said application based upon permissions data associated with said application.

37. The process of claim 36 wherein said determining step includes the steps of:

providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards from which said application may be deleted;

comparing said personalization data with said permissions data; and

deleting said application from said IC card provided said personalization data falls within said set of cards from which said application may be deleted.

38. The process of claim 37, wherein said personalization data comprises data representative of a card identification designation.

39. The process of claim 37 or claim 38, wherein said personalization data comprises data representative of a card issuer.

40. The process of any of claims 37 to claim 39, wherein said personalization data comprises data representative of a product class.

41. The process of any of claims 37 to claim 40, wherein said personalization data further comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

42. An IC card system comprising at least one IC card, an application to be deleted from said card and means for determining whether said card is qualified to delete said application from said card.

43. The IC card system of claim 42, wherein said IC card contains card personalization data, and said application is assigned application permissions data set representing at least one set of IC cards from which said application may be deleted.

44. The IC card system of claim 43, wherein said determining means compares said card personalization data with said application permissions data.

45. The IC card system of claim 44, wherein whether said application is deleted from said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is deleted.

ABSTRACT

Multi-Application IC Card System

A multi-application IC card system is disclosed having selective application loading and deleting capability. Prior to loading an application onto an IC card a test is conducted to determine if the card is qualified to receive the application using personalization data stored on the card and comparing it with permissions data associated with the application indicating one or more sets of cards upon which the application may be loaded. If the personalization data of the card falls within the allowable set of permissions for that application then the card may be loaded with the application. Preferably, the personalization data includes data representative of the card number, issuer, a product class and the date on which the card is personalized.

ANNEX B TO THE DESCRIPTION

ANNEX B

KEY TRANSFORMATION UNIT FOR AN IC CARD

ANNEX B TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two

different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and

entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet. At least three issues need to be addressed in a system which provides such a capability.

The first issue is to make sure that the IC card receiving the application is the intended IC card and not another IC card. The second issue is determining how the IC card can authenticate that the application came from the proper application provider and not an unknown third party. The third issue concerns preventing third parties from reading the application and making an unauthorized copy. If a portion of the application is encrypted to address the latter issue, the intended IC card needs to have access to the correct key to decrypt the application. In a system with many IC cards and additionally many application

ANNEX B TO THE DESCRIPTION

providers, a secure key transfer technique is required so that the intended IC card can use the correct key for the application which is received. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of this invention to provide a key transfer and authentication technique and specifically to provide a secure IC-card system that allows for the secure transfer of smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card system and method for securely loading an application onto an IC card including providing a secret and public key pair for the IC card, encrypting at least a portion of the application using a transfer key, encrypting the transfer key using the IC card's public key to form a key transformation unit, transmitting the encrypted application and the key transformation unit to the IC card, decrypting the key transformation unit using the IC card's secret key to provide the transfer key, decrypting the encrypted application using the provided transfer key and storing the decrypted application on the IC card.

In a preferred embodiment, the secure loading system and method allows the application provider to encrypt two or more portions of the application to be transmitted with two or more different keys, encrypt the two or more keys with the public key of the IC card to form a key transformation unit including the

ANNEX 6 TO THE DESCRIPTION

locations of the encrypted portions. Both the encrypted application and the key transformation unit are sent to the IC card. Because the decryption keys are encrypted with the IC card's public key, only the IC card's secret key can decrypt the key transformation unit. The transfer keys and the locations of the encrypted portions are recovered from the decrypted key transformation unit and the application is decrypted using the recovered transfer keys. This ensures that only the intended IC card can decrypt and use the application which was transmitted to that IC card.

In a preferred embodiment, an application load certificate is also sent to the IC card which is receiving the application. The application load certificate contains the public key of the application provider encrypted by the secret key of the certificate authority ("CA"), or the entity that manages the overall security of the IC card system. The IC card then uses a certificate authority public key to make sure that the certificate was valid by attempting to verify the application load certificate with the CA's public key. The IC card then uses the recovered application provider's public key to verify that the application provider was in fact the originator of the application by verifying the sent application signature generated with the application provider's corresponding secret key.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

ANNEX B TO THE DESCRIPTION

Fig. 1 is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection

with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features.

The flexibility of loading applications at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable

ANNEX 6 TO THE DESCRIPTION

lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. One solution to this problem is addressed in a related application entitled "Secure Multi-Application IC Card System Having Selective Loading and Deleting Capability" by Everett et al., filed February 12, 1998 and assigned to Mondex International, which is hereby incorporated by reference. Two additional security concerns also need to be addressed when loading an application from a remote source, or even from a local terminal, onto an IC card. First, the source of the application must be authenticated as the proper originator so that applications which may contain viruses or simply take up the limited storage memory in an IC card are not allowed to be loaded onto an IC card. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so other people cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key.

ANNEX 6 TO THE DESCRIPTION

After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a related decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to decipher the message. If the message was readable after the public key was applied to it, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and

ANNEX B TO THE DESCRIPTION

encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security and is more robust than symmetric encryption. However, asymmetric encryption is more processor costly that symmetric encryption. A example of an asymmetric encryption method is RSA.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with symmetric key 1, decodes the data using symmetric key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1 shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security

ANNEX B TO THE DESCRIPTION

measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The Certificate Authority 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets provided to them. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored

ANNEX B TO THE DESCRIPTION

on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is signed with a public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be

ANNEX B TO THE DESCRIPTION

written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetrical key and the

ANNEX 6 TO THE DESCRIPTION

conventionally known DES algorithm to transform the data. The data can later be recovered by applying the key to the known DES algorithm. Encrypted location 311 shows a second portion of the application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

One of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's

ANNEX B TO THE DESCRIPTION

secret key which is known only to the card, the card's public key which is stored on the card and the card's public key certificate which is the card's public key signed by one of the CA's secret keys. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by encrypting the transport public key mkd_pk with the secret key of the CA, indicated as follows:

$$mkd\_pk\_c = [mkd\_pk]_{CA\_sk}$$

ANNEX B TO THE DESCRIPTION

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk. The decrypted public key certificate can then be compared with the public key to verify that the key certificate was certified (signed) by the CA.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key

ANNEX 6 TO THE DESCRIPTION

of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By signing the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys provided to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies.

No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could be 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on

ANNEX 6 TO THE DESCRIPTION

the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring of the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider.

ANNEX 6 TO THE DESCRIPTION

The IC card, when it receives the ALC 209 as part of the ALU 201, will open the ALC 209 with the public key of the CA. This ensures that the CA signed the application load certificate and that it is genuine. After decrypting the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to decrypt the signed AU 205 in order to verify that AU 203 was signed by the application provider. AU signed 205 is verified with the Application Provider Public Key 801. The recovered AU 803 is then compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information because the application provider's public key is provided to it as part of the application load certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application

ANNEX B TO THE DESCRIPTION

provider, (2) being loaded on the intended card and (3) certified by the CA. The ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in the EEPROM of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 decrypts the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key successfully verifies the ALC 209, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC successfully, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the application load certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the

process continues.

Step 907 uses the application providers public key which was recovered from the verified ALC to verify the AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signed 205. If the ALU is successfully verified, then it is accepted as having been generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

ANNEX 6 TO THE DESCRIPTION

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its static memory once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

ANNEX B TO THE DESCRIPTION

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic unit 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like performing

ANNEX 6 TO THE DESCRIPTION

cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the integrated circuit chip for the application provider and for the certification authority. CPU 1101 present in the IC chip for the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 at the certification authority is used to sign the Application Load Certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the

ANNEX 6 TO THE DESCRIPTION

same secure loading process can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

ANNEX 6 TO THE DESCRIPTION

I CLAIM:

1. A method for securely loading an application onto an IC card comprising the steps of:

providing a secret key and public key pair for said IC card;

encrypting at least a portion of said application using a transfer key;

encrypting said transfer key using said IC card's public key to form a key transformation unit;

transmitting said encrypted application and said key transformation unit to said IC card;

decrypting said key transformation unit using said IC card's secret key to recover said transfer key; and

decrypting said encrypted application using said recovered transfer key.

2. The method of claim 1, further including the step of storing said decrypted application on said IC card.

3. The method of claim 1, wherein said encryption technique using said transfer key transfer key is symmetric.

4. The method of claim 3, wherein said symmetric technique is DES.

ANNEX 6 TO THE DESCRIPTION

5. The method of claim 1, wherein said IC card's public and private keys are provided using an asymmetric technique.

6. The method of claim 5, wherein said asymmetric technique is RSA.

7. The method of claim 1, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

8. The method of claim 1, further including the steps of enciphering a second portion of said application exclusive of said at least a portion of said application.

9. The method of claim 8, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

10. The method of claim 8, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

11. The method of claim 8, wherein said key transformation unit indicates the location of said second portion of said application.

ANNEX B TO THE DESCRIPTION

12. The method of claim 1, wherein said key transformation unit indicates the location of said at least a portion of said application.

13. The method of claim 1, wherein said key transformation unit indicates the number of encrypted portions of said application.

14. The method of claim 1, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; encrypting said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed application and transmitting said signed application and said application load certificate to said IC card.

15. The method of claim 14, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

16. The method of claim 15, further including the steps of verifying the signed encrypted application using the application provider's public key from said decrypted application load certificate.

ANNEX B TO THE DESCRIPTION

17. The method of claim 16, wherein said verified application signature is compared to sent encrypted application to determine if they are equivalent.

18. An IC card system comprising:

at least one IC card;

an application provider for providing an application to said at least one IC card;

a communications link coupled to said at least one IC card and said application provider;

a public key and secret key set generated for said IC card;

a transport key generated for use by said applications provider; and

an application, wherein at least a portion of said application is encrypted by said application provider using said transport key; said transport key is encrypted using said IC card's public key to form a key transformation unit; wherein said encrypted application and said key transformation unit are then transmitted to said IC card over said communications link; said transmitted key transformation unit is decrypted using said IC card's private key to recover said transport key; and said transmitted application is decrypted using said recovered transport key to recover said application.

19. The system of claim 18, wherein said recovered application is stored on said card.

ANNEX 6 TO THE DESCRIPTION

20. The system of claim 18, wherein said encryption technique using said transfer key transfer key is symmetric.

21. The system of claim 20, wherein said symmetric technique is DES.

22. The system of claim 18, wherein said IC card's public and private keys are provided using an asymmetric technique.

23. The system of claim 22, wherein said asymmetric technique is RSA.

24. The system of claim 18, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

25. The system of claim 18, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

26. The system of claim 25, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

27. The system of claim 25, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

ANNEX B TO THE DESCRIPTION

28. The system of claim 25, wherein said key transformation unit indicates the location of said second portion of said application.

29. The system of claim 18, wherein said key transformation unit indicates the location of at least a portion of said application.

30. The system of claim 18, wherein said key transformation unit indicates the number of encrypted portions of said application.

31. The system of claim 18, further including a certification authority, wherein a public key and secret key set is provided for an application provider; a public and secret key set is provided for said certification authority; said certificate authority's secret key is used to sign said application provider's public key to produce an application load certificate; said application provider's secret key is used to further sign said encrypted application to produce a signed encrypted application and said signed encrypted application and said application load certificate is transmitted to said IC card.

32. The system of claim 31, wherein the IC card verifies said application load certificate with said certification authority's public key.

ANNEX B TO THE DESCRIPTION

33. The system of claim 32, wherein said IC card verifies the signed encrypted application using the application provider's public key from said verified application load certificate.

34. The system of claim 33, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

35. A method for transmitting data in a secure manner from a first microprocessor based device to a second microprocessor based device, comprising the steps of:

encrypting at least a portion of said data at said first device using a transfer key;

encrypting said transfer key with a second key at said first device to form a key transformation unit;

transmitting said encrypted data and said key transformation unit to said second device;

decrypting said key transformation unit at said second device to recover said transfer key; and

decrypting said encrypted data using said recovered transfer key.

36. The method of claim 35, further including the step of storing said decrypted data in said second device.

37. The method of claim 35, wherein said second key is from a public key and private key set used in asymmetric encryption.

38. The method of claim 35, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

39. The method of claim 35, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

40. The method of claim 39, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

41. The method of claim 39, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

42. The method of claim 39, wherein said key transformation unit indicates the location of said second portion of said application.

43. The method of claim 35, wherein said key transformation unit indicates the location of said at least a portion of said application.

44. The method of claim 35, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authority's secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting said signed application and said application load certificate to said IC card.

45. A method for processing a data transmission comprising the steps of:

receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

decrypting said key transformation unit to recover said first key;

decrypting said encrypted application using said first key; and

storing said decrypted application.

46. The method of claim 45, wherein said second key is from a public key and private key set used in asymmetric encryption.

47. The method of claim 45, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

ANNEX 6 TO THE DESCRIPTION

48. The method of claim 45, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

49. The method of claim 48, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

50. The method of claim 48, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

51. The method of claim 48, wherein said key transformation unit indicates the location of said second portion of said application.

52. The method of claim 45, wherein said key transformation unit indicates the location of said at least a portion of said application.

53. The method of claim 45, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further encrypting said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting

ANNEX B TO THE DESCRIPTION

said signed application and said application load certificate to said IC card.

54. The method of claim 53, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

55. The method of claim 54, further including the steps of verifying the signed encrypted application using the application provider's public key from said verified application load certificate.

56. The method of claim 55, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

57. An apparatus for processing a data transmission comprising the steps of:

means for receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

means for decrypting said key transformation unit to recover said first key;

means for decrypting said encrypted application using said first key; and

means for storing said decrypted application.

ANNEX B TO THE DESCRIPTION

58. The apparatus of claim 57, wherein said second key is from a public key and private key set used in asymmetric encryption.

59. The apparatus of claim 57, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

60. The apparatus of claim 57, further including means for enciphering a second portion of said application exclusive of said at least a portion of said application.

61. The apparatus of claim 60, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

62. The apparatus of claim 60, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

63. The apparatus of claim 60, wherein said key transformation unit indicates the location of said second portion of said application.

64. The apparatus of claim 57, wherein said key transformation unit indicates the location of said at least a portion of said application.

ANNEX B TO THE DESCRIPTION

65. The apparatus of claim 60, further including means for verifying an application load certificate with said certification authority's public key.

66. The apparatus of claim 65, further including means for verifying the signed encrypted application using an application provider's public key located in said verified application load certificate.

67. The apparatus of claim 66, wherein said verified application signature is compared to the said encrypted application to determine if they are equivalent.

ANNEX B TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

A multi-application IC card system and method is disclosed providing a secure data transmission technique. The method is used, for example, to load an application from an application provider, which could be remote, to an IC card. At least a portion of the application is encrypted using a transfer key. The transfer key is then encrypted using the public key of a public/secret key pair of the intended IC card to form a key transformation unit. The encrypted application and key transformation unit are then sent to the IC card and the IC card decrypts the key transformation unit using its secret key. The transfer key is then recovered and used to decrypt the encrypted application. The application can then by stored on the IC card and accessed by the card user.

ANNEX C TO THE DESCRIPTION

ANNEX C

IC CARD TRANSPORTATION KEY SET

ANNEX C TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite

ANNEX C TO THE DESCRIPTION

cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because

these data transmission lines are not typically secure lines, a number of security and entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his or her IC card into a home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet.

An entity which transmits an application or data to an IC card requires that only the intended IC card should receive the transmitted data. Third parties should not be able to intercept and view the data. Additionally, a transmitting entity will require verification that the IC card which has requested information is actually part of the overall IC card system and not simply posing as being part of the system. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of this invention to provide a secure transfer technique and specifically to provide a secure IC-card system that allows for the secure transfer of data including smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card method and apparatus for securely transporting data including an application onto an IC card including storing a secret and public key pair on the IC card, retrieving the stored public key from the IC card, encrypting at least a portion of the data to be transported using the public key, transmitting the encrypted data to the IC card and decrypting the encrypted data using the IC card's secret key.

In a preferred embodiment, a certification authority ("CA") or the entity that manages the overall security of the IC card system, encrypts (or digitally signs) a copy of the IC card's public key and the signed copy is also stored on the IC card. The entity transmitting the data to the IC card can verify that the CA has approved the card by retrieving using the IC card's signed public key and verifying the signed public key using the public key of the CA. If verification is successful, the entity has verified that the CA approved the IC card.

ANNEX C TO THE DESCRIPTION

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1A is a block diagram of the secure data transfer system which securely transfers data from a transferring entity to an IC card.

Fig. 1B is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing

ANNEX C TO THE DESCRIPTION

the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the

ANNEX C TO THE DESCRIPTION

card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features. Additionally, an IC card needs to receive data regarding personal information such as new credit card account numbers or updated information.

The flexibility of loading applications and transmitting data at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications and data both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications and data onto an IC card, the application provider needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so entities other than the IC card cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical

ANNEX C TO THE DESCRIPTION

formula which encrypts data by transforming the data using the formula and key. After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to verify the message. If the message was verified with the public key, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the

ANNEX C TO THE DESCRIPTION

data. If a combination of keys is used, a person could both authenticate and encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security. However, asymmetric encryption is relatively processor costly (processor cost is associated with computation time) compared with symmetric encryption. An example of asymmetric encryption method is RSA®.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with key 1, decodes the data using key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1A shows a block diagram of the entities used in transporting data in a secure manner in an IC card system. The transmitting entity 1 can be a card issuer, bank, IC card or other entity which desires to transport data to an IC card 3. The transmitting entity 1 preferably initiates the data transfer process. Alternatively, the IC card 3 can initiate the data transfer process if the card requires data from the transmitting entity 1.

The transmitting entity 1 is connected to interface device 5 (e.g., a terminal that communicates with an IC card). Data conduit 7 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. In this example, the transmitting entity 1, which is remotely located from IC

ANNEX C TO THE DESCRIPTION

card 3, desires to send data in a secure manner to the IC card. However, because the data link is an "open" link (i.e. not a private link) and subject to third parties possibly intercepting or replacing data being transmitted, security measures are needed to guarantee that only the intended IC card will receive the transmitted data. The Certificate Authority 9 can also be used to authenticate that the IC card has been validated as part of the IC card system.

In Figure 1A, a private (or secret) key 19 and corresponding public key 15 is generated for IC card 3. The keys are preferably generated using an asymmetric encryption algorithm such as RSA®. The keys can be generated at the CA 9 or any other location because they are specific only to the IC card 3 and no other copies need to be kept. A third data item, the public key certificate 17, is also generated and stored on the IC card 3.

The public key certificate 17 is generated by signing the public key 15 with the private key of the CA 9. This allows a person with the public key of the CA 9 to verify that the CA digitally signed the IC card's public key in order to certify the IC card's individual key set. The public key certificate can be generated by the CA at the time the IC card private/public key set is generated or at a subsequent time.

When a data transfer is initiated by the transmitting entity 1, the IC card 3 is contacted through the interface device 5 and the IC card 3 sends its public key 15 and its public key certificate 17 to the transmitting entity 1. The transmitting entity then verifies the public key certificate with public key of the CA 13 (which is publicly available from the CA 9 and may be stored in the transmitting

entity 1) thus determining if the CA 9 digitally signed the public key and verifying that the IC card is a valid card.

The transmitting entity 1 then encrypts the data to be transmitted with the IC card's public key. The transmitting entity 1 then transmits the encrypted data 11 to the interface device 5 and to the IC card 3. The IC card 3 decrypts the encrypted data with its corresponding private (also called secret) key 19. The data can then be processed by the IC card 3. Only the IC card 3 has a copy of its private key so only the intended IC card can access the encrypted data. This ensures that third parties cannot access the encrypted data and correspondingly that only the intended IC card will be able to read and process the data.

Figure 1B shows a secure method for loading applications onto an IC card. Figure 1B shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. IC card 103 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security measures which authenticate the application itself, the application provider and the IC card must be used to ensure

ANNEX C TO THE DESCRIPTION

the integrity of the system. The CA 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1B, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

ANNEX C TO THE DESCRIPTION

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is encrypted with the public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in

ANNEX C TO THE DESCRIPTION

MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetric key and the conventionally known DES-based algorithm to transform the data. The data can later be recovered by applying the key to a conventionally known DES-based decryption algorithm. Encrypted location 311 shows a second portion of the

ANNEX C TO THE DESCRIPTION

application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys and decryption algorithm. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

In accordance with the present invention, one of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's secret key which is known only to the card, the card's public key which is stored on the card and the card's public key

ANNEX C TO THE DESCRIPTION

certificate which is the card's public key signed by the CA's secret key. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA from a standard asymmetric encryption technique such as RSA® and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. As with the secret key, once the public key is stored on the card, the CA (or other key provider) deletes from its systems the public key data so that the only copy of the public key is kept in the card. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by signing the transport public key mkd_pk with the secret key of the CA, indicated as follows:

ANNEX C TO THE DESCRIPTION

$$mkd\_pk\_c = [mkd\_pk]_{CA\_sk}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key of an IC card is freely available to anyone and can be obtained directly from the

ANNEX C TO THE DESCRIPTION

card or from the CA. By encrypting the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies.

No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could by 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with

ANNEX C TO THE DESCRIPTION

the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring to the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider. The IC card, when it receives the ALC 209 as part of the ALU 201, will verify the

ANNEX C TO THE DESCRIPTION

ALC 209 with the public key of the CA. This ensures that the CA signed the Application Load Certificate and that it is genuine. After verifying the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to verify the signature of the AU 205 in order to verify that AU 203 was signed by the application provider. AU signature 205 is verified with the Application Provider Public Key 801 and compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information efficiently because the application provider's public key is provided to it as part of the Application Load Certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application provider, (2) being loaded on the intended card and (3) certified by the CA. The

ANNEX C TO THE DESCRIPTION

ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in an I/O buffer of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 verifies the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key verifies the ALC 209 properly, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC properly, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the Application Load Certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the process continues.

ANNEX C TO THE DESCRIPTION

Step 907 uses the application providers public key which was recovered from the verified ALC to verify AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key to authenticate that the application was provided by the correct application provider. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signature 205. If the two data blocks match, then the ALU is verified as being generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

ANNEX C TO THE DESCRIPTION

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its EEPROM once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

ANNEX C TO THE DESCRIPTION

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like

ANNEX C TO THE DESCRIPTION

cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The IC card private key will be stored in a secure memory location. The IC card public key and public key certificate is preferably stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the application provider, transmitting entity and for the CA. CPU 1101 present in the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 present in the certification authority is used to sign the Application Load Certificate and the public key certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of

the invention.

For example, while loading an application is discussed herein, the same secure loading processes can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

ANNEX C TO THE DESCRIPTION

WE CLAIM:

1. A method for securely transporting data onto an integrated circuit card by using an individualized key set for said card, comprising the steps of:

storing a private key and public key pair unique to said integrated circuit card in said memory located on said integrated circuit card;

retrieving said stored public key from said integrated circuit card;

encrypting at least a portion of said data to be transported onto said card, using said retrieved public key;

transmitting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said integrated circuit card's private key to recover said transported data.

2. The method of claim 1, further including the step of storing said decrypted data on said integrated circuit card.

3. The method of claim 1, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

ANNEX C TO THE DESCRIPTION

4. The method of claim 3, wherein said public key certificate is verified with said certification authority's stored public key prior to said transmitting steps.

5. The method of claim 4, wherein said retrieved public key certificate is recovered and compared with said stored public key.

6. The method of claim 5, wherein said integrated circuit card's public and private keys are provided using an asymmetric technique.

7. The method of claim 6, wherein said asymmetric technique is RSA.

8. A method performed by an integrated circuit card for processing incoming data transmission to said integrated circuit card by using an individualized key set for the card, comprising the steps of:

receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

retrieving a unique private key for said integrated circuit card which is part of said individualized key set; and

decrypting said encrypted data with said unique private key to recover said data.

ANNEX C TO THE DESCRIPTION

9. The method of claim 8, further including the step of storing said decrypted data on said integrated circuit card.

10. The method of claim 8, wherein said individualized key set is generated by asymmetric encryption.

11. The method of claim 8, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

12. The method of claim 11, wherein said public key certificate is retrieved prior to said transmitting steps.

13. The method of claim 12, wherein said retrieved public key certificate is verified using said certification authority's stored public key.

14. An apparatus located on an integrated circuit card by using an individualized key set for said card for processing an incoming secure data transmission comprising:

means for receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

ANNEX C TO THE DESCRIPTION

means for retrieving a unique public key for said integrated circuit card which is part of said individualized key set; and

means for decrypting said encrypted data with said unique private key to recover said data.

15. The apparatus of claim 14, further comprising means for storing said data on said integrated circuit card.

16. The apparatus of claim 14, further including means for retrieving a public key certificate which is generated by a certificate authority digitally signing said unique public key.

17. The apparatus of claim 16, further including means for transmitting said public key certificate prior to said receiving means receiving.

18. The apparatus of claim 17, wherein said transmitted public key certificate is verified using said certification authority's stored public key.

19. A method of securely transporting data onto an integrated circuit card by using an individualized key set for the card, comprising the steps of:

providing a first unique private and public key pair for a certification authority;

storing a second unique private and public key pair which

ANNEX C TO THE DESCRIPTION

form said individualized key set for said integrated circuit card in a memory located on said integrated circuit card;

encrypting said second public key with said first certification authority's private key to form a public key certificate;

storing said public key certificate on said integrated circuit card;

retrieving said stored public key certificate from said integrated circuit card;

verifying said public key certificate with said first public key to ensure that said public key certificate is valid;

encrypting at least a portion of said data using said retrieved second public key;

transporting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said second private key to retrieve said data.

20. The method of claim 19, wherein said data comprises an application.

ANNEX C TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

Method and apparatus for securely transporting data onto an IC card. The method is used, for example, to transport data, including application programs, in a secure manner from a source located outside the IC card. At least a portion of the data is encrypted using the public key of a public/secret key pair of the intended IC card unit. The encrypted data is then sent to the IC card and the IC card verifies the key transformation unit using its unique secret key. The data can then be stored on the IC card. A copy of the public key signed by a certification authority can be used to verify that the card is authorized to be part of the overall authorized system.

CLAIMS

I CLAIM:

1. A method of loading an application copy onto an integrated circuit card, wherein said application copy is one of a plurality of copies of an application, said application copy having an associated application identifier that uniquely identifies said application from other applications and an application copy number that is unique for each copy of said application, said integrated circuit card comprising a microprocessor and memory coupled to said microprocessor, said memory comprising an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto said integrated circuit card, said method comprising:

receiving by said integrated circuit card said application copy, said application identifier, and said application copy number;

determining by said integrated circuit card whether said application identifier and said application copy number are contained in said application history list area; and

failing to load said application copy by said integrated circuit card if said application identifier and said application copy number are contained in said application history list area.

2. The method of claim 1, further comprising the steps of:

allocating a predetermined portion of said memory for said application history list area;

determining by said integrated circuit card whether said application history list area is full; and

failing to load said application copy if said application history list is full.

3. The method of claim 1 or claim 2, further comprising the step of:

adding said application identifier and said application copy number to said application history list area if said application identifier and said application copy number are not contained in said application history list area.

4. The method of claim 1 or claim 2, further including the step of:

adding said application identifier and said application copy number to said application history list area if said application identifier and said application copy number are not contained in said application history list area and said application copy number is not zero.

5. The method of any preceding claim, wherein said application copy comprises application code and application data and a portion of said application data comprises units of value that may be exchanged for goods or services.

6. The method of any preceding claim, wherein said application copy comprises application code and application data and wherein said application identifier and said application copy number are contained in said application data.

7. The method of any preceding claim, further comprising the step of:

transmitting said application copy, said application identifier, and said application copy number to said integrated circuit card by an application provider.

8. The method of claim 7, further comprising the step of:

encrypting by said application provider at least a portion of said application copy before transmitting said application copy to said integrated circuit card.

9. The method of claim 8, further comprising the step of:

transmitting by said application provider a key transformation unit comprising information relating to the encryption of said portion of said application copy.

10. The method of claim 9, wherein said integrated circuit card has a first public key pair, and further comprising the step of:

encrypting said key transformation unit by said application provider with the public key of said first public key pair before transmitting said key transformation unit to said integrated circuit card.

11. The method of claim 10, further comprising the steps of:

decrypting by said integrated circuit card said encrypted key transformation unit with the secret key of said first public key pair; and

decrypting said application copy using the information contained in said decrypted key transformation unit.

12. The method of claim 7 or any claim dependent thereon, wherein said application provider has a second public key pair, and further comprising the steps of:

forming a signed application copy by said application provider by encrypting said application copy with the secret key of said second public key pair; and

transmitting by said application provider said signed application copy to said integrated circuit card.

13. The method of claim 12, further comprising the steps of:

registering the public key of said second public key pair with a certification authority, which has a third public key pair.

providing a certificate by said certification authority to said application provider by encrypting the public key of said second public key pair with the secret key of said third public key pair; and

transmitting said certificate by said application provider to said integrated circuit card.

14. The method of claim 13, further comprising the steps of:

obtaining the public key of said second key pair by said integrated circuit card by decrypting said certificate using the public key of said third public key pair;

verifying by said integrated circuit card said signed application copy using the public key of said second public key pair;

failing to load said application copy by said integrated circuit card if said signed application copy is not verified.

15. An integrated circuit card, comprising:

a microprocessor;

a memory coupled to said microprocessor, said memory including an application history list area for storing application identifiers and application copy numbers, each application identifier and each application copy number being associated with an application copy, said application copy being one of a plurality of copies of an application, each application identifier uniquely identifying an application from other applications, and each application copy number uniquely identifying an application copy from other application copies;

means for determining whether an application identifier and an application copy number associated with an application copy to be loaded into said memory area are contained in said application history list area; and

means for failing to load said application copy to be loaded if said associated application identifier and said associated application copy number are contained in said application history list area.

16. The integrated circuit card of claim 15, wherein said application history list area is an allocated, predetermined portion of said memory, and further comprising:

means for determining whether said application history list area is full; and

means for failing to load said application copy to be loaded if said application history list area is full.

17. The integrated circuit card of claim 15 or claim 16, further comprising means for adding said associated application identifier and said associated application copy number of said application copy to be loaded into said application history list area if said application identifier and said application copy number are not contained in said application history list area.

18. The integrated circuit card of any of claims 15 to 17, further comprising means for adding said associated application identifier and said associated application copy number of said application copy to be loaded into said application history list area if said application identifier and said application copy number are not contained in said application history list area and said application copy number is not zero.

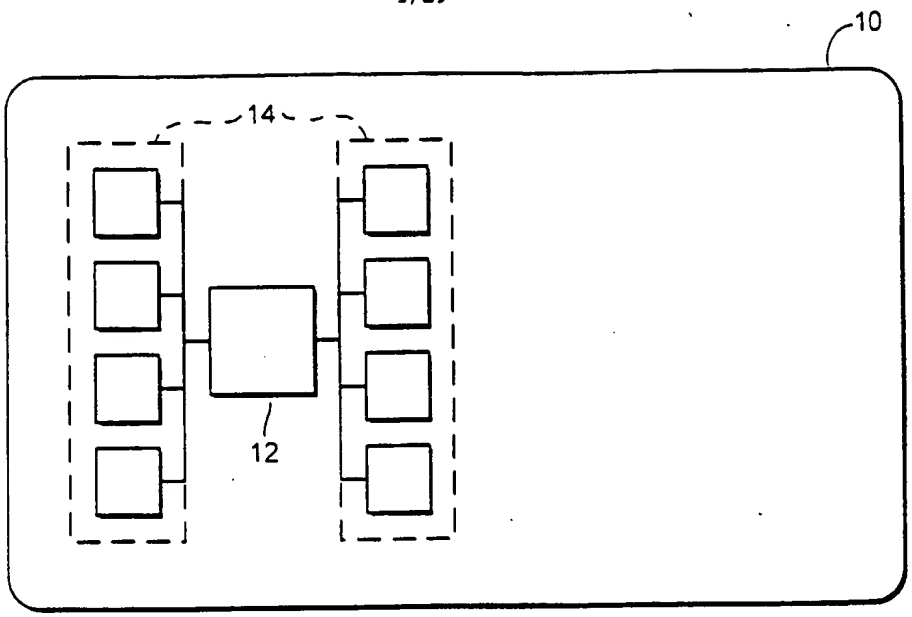


FIG. 1

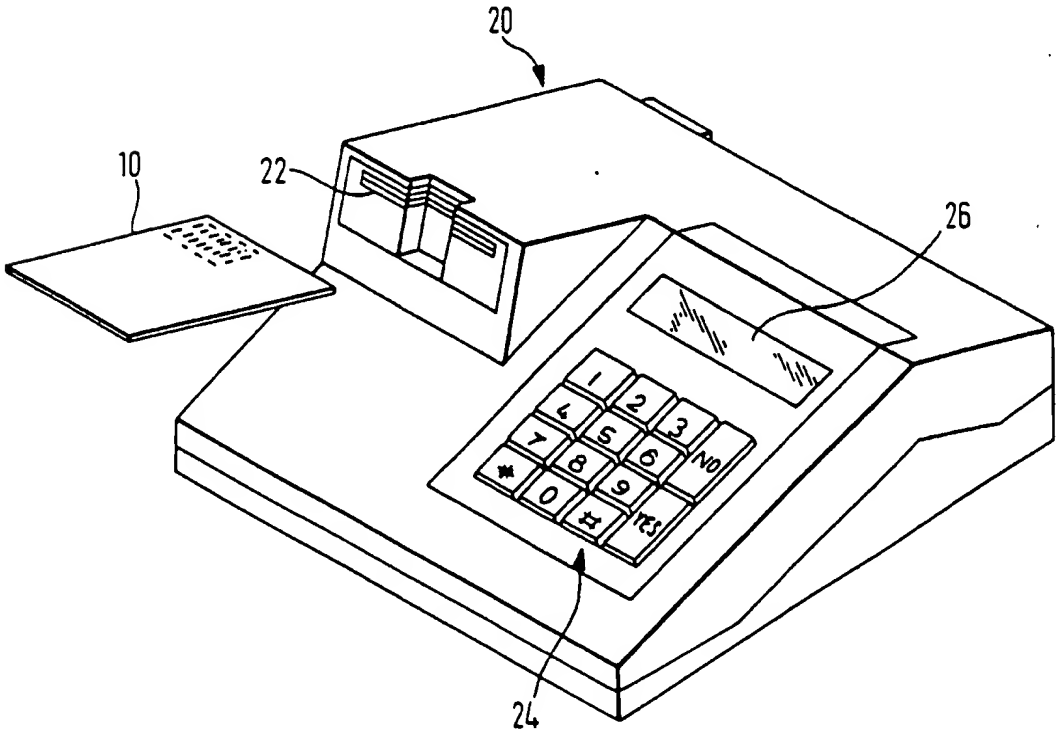


FIG. 2

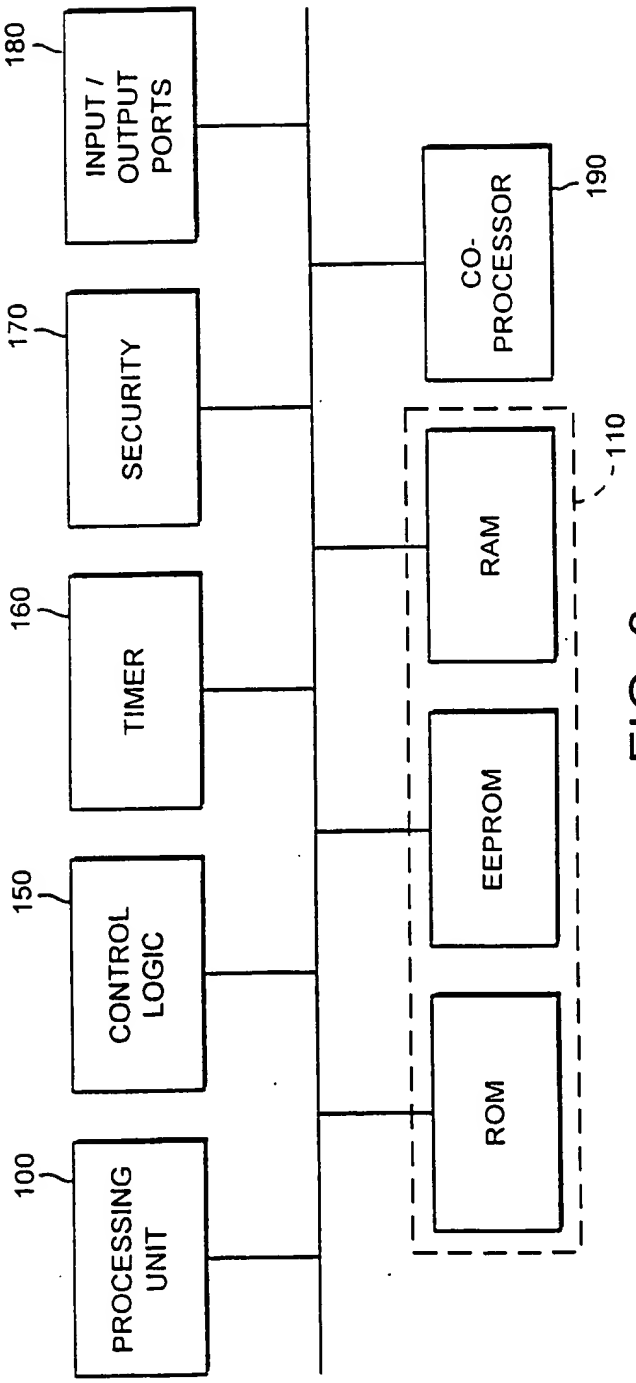

180
INPUT / OUTPUT PORTS
170
SECURITY
160
TIMER
150
CONTROL LOGIC
100
PROCESSING UNIT
190
CO-PROCESSOR
110
RAM
EEPROM
ROM

FIG. 3

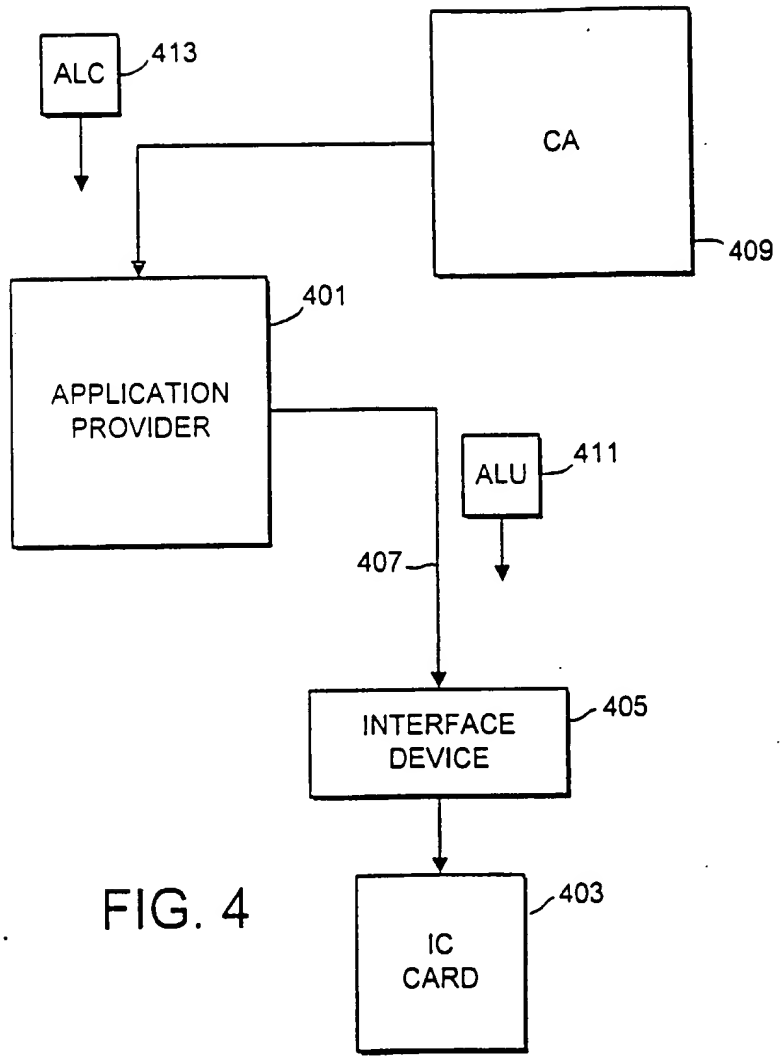

ALC
413
CA
409
401
APPLICATION PROVIDER
ALU
411
407
INTERFACE DEVICE
405
IC CARD
403

FIG. 4

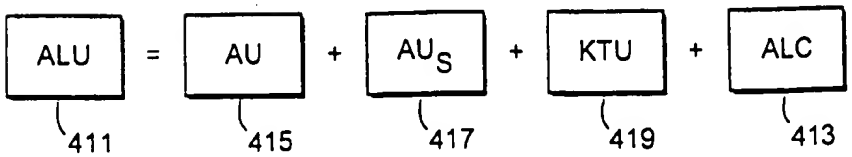

ALU
=
AU
+
AUS
+
KTU
+
ALC
411
415
417
419
413

FIG. 5

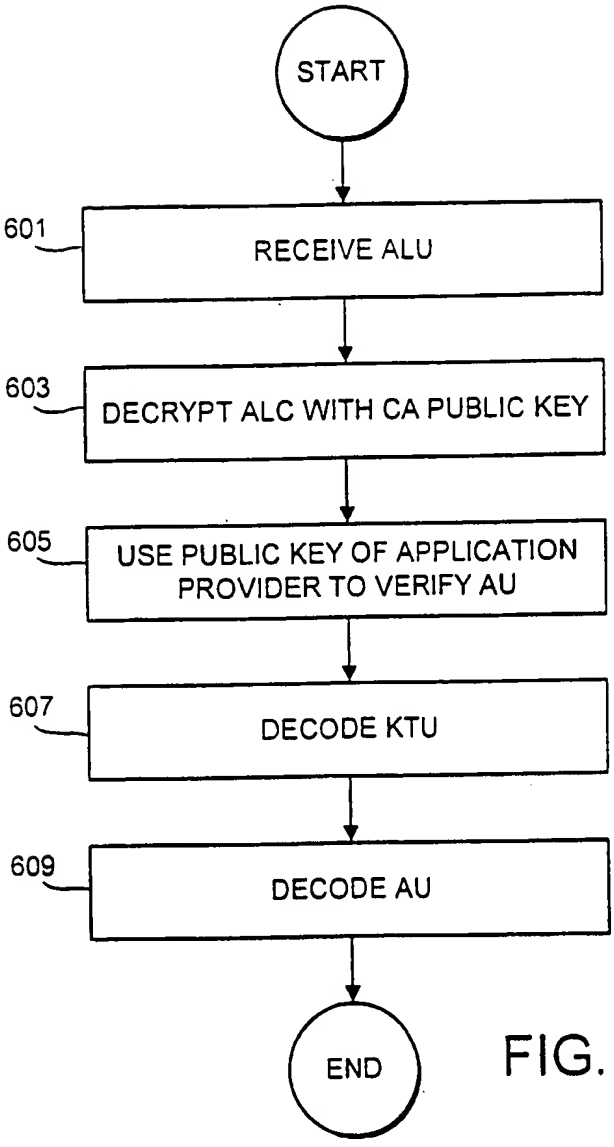

START
601
RECEIVE ALU
603
DECRYPT ALC WITH CA PUBLIC KEY
605
USE PUBLIC KEY OF APPLICATION PROVIDER TO VERIFY AU
607
DECODE KTU
609
DECODE AU
END

FIG. 6

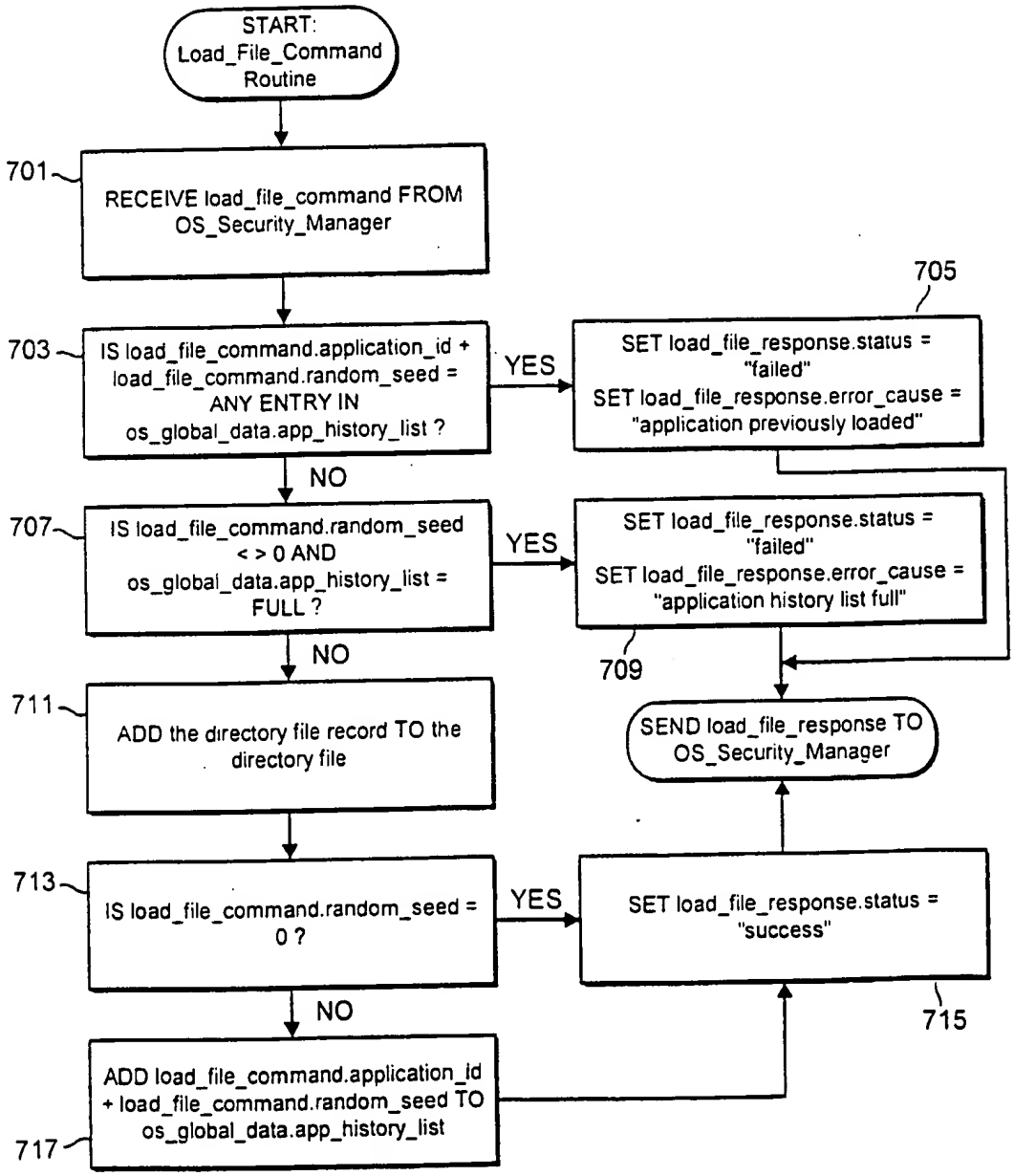

START: Load_File_Command Routine
701
RECEIVE load_file_command FROM OS_Security_Manager
703
IS load_file_command.application_id + load_file_command.random_seed = ANY ENTRY IN os_global_data.app_history_list ?
YES
705
SET load_file_response.status = "failed"
SET load_file_response.error_cause = "application previously loaded"
NO
707
IS load_file_command.random_seed < > 0 AND os_global_data.app_history_list = FULL ?
YES
SET load_file_response.status = "failed"
SET load_file_response.error_cause = "application history list full"
709
NO
711
ADD the directory file record TO the directory file
SEND load_file_response TO OS_Security_Manager
713
IS load_file_command.random_seed = 0 ?
YES
SET load_file_response.status = "success"
715
NO
ADD load_file_command.application_id + load_file_command.random_seed TO os_global_data.app_history_list
717

FIG. 7

ANNEX A TO THE DRAWINGS

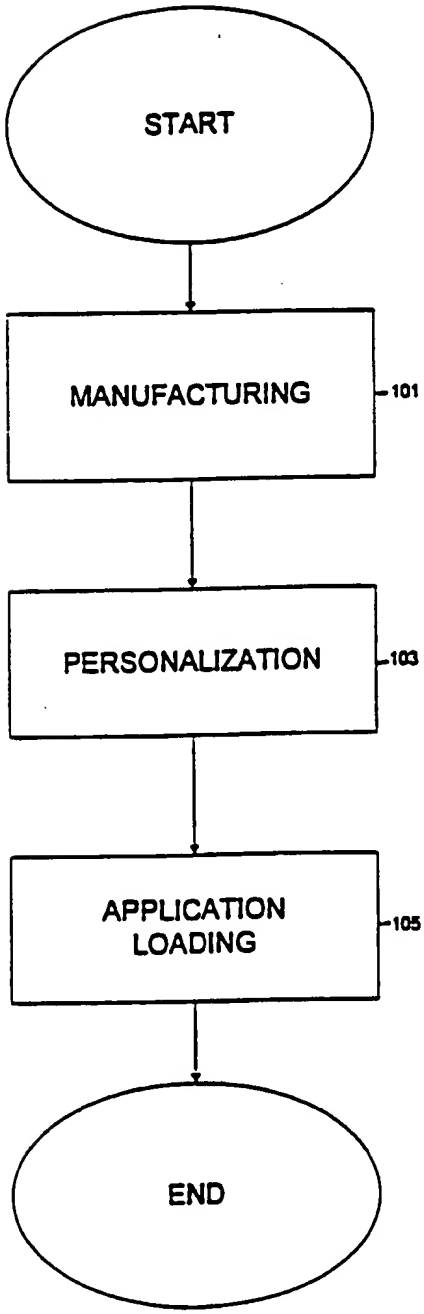


FIG. 1

ANNEX A TO THE DRAWINGS

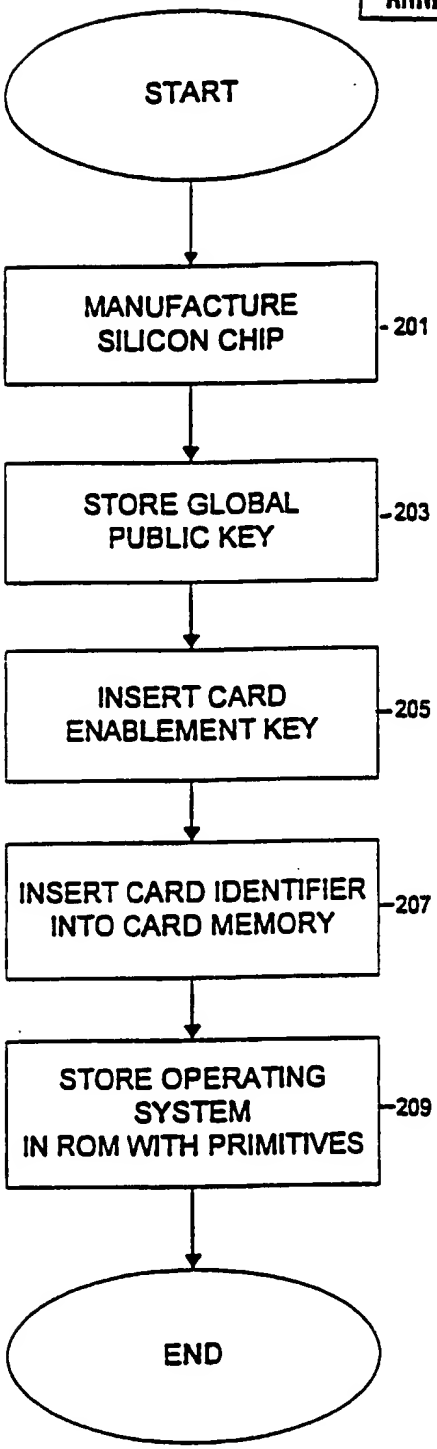


FIG. 2

ANNEX A TO THE DRAWINGS

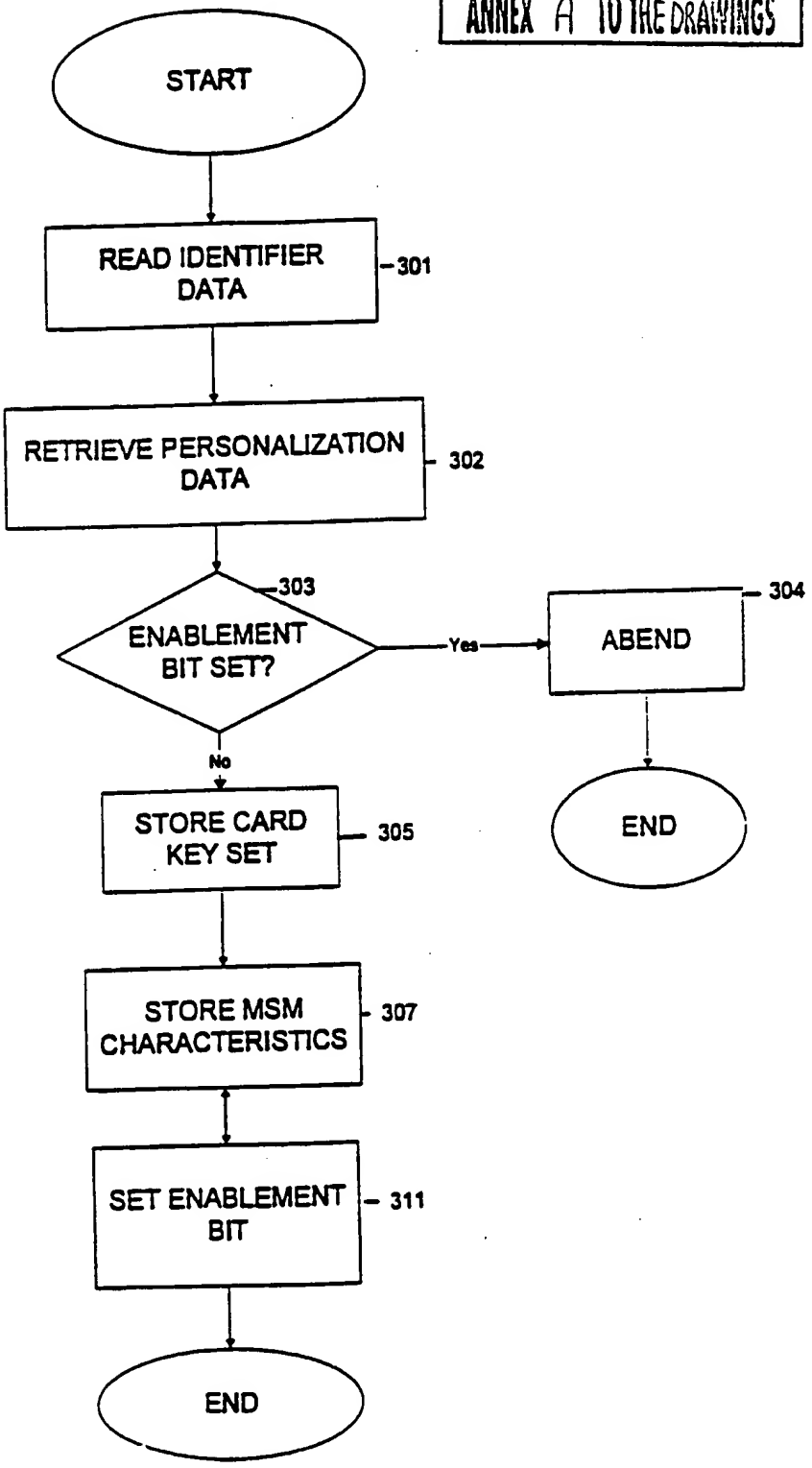


FIG. 3

ANNEX A TO THE DRAWINGS

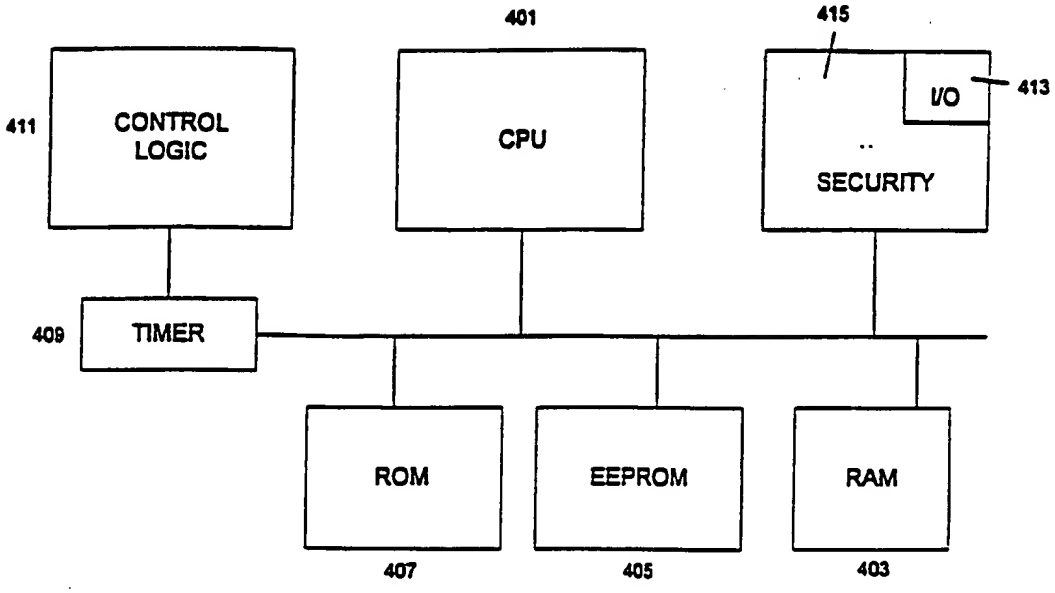


FIG. 4

ANNEX A TO THE DRAWINGS

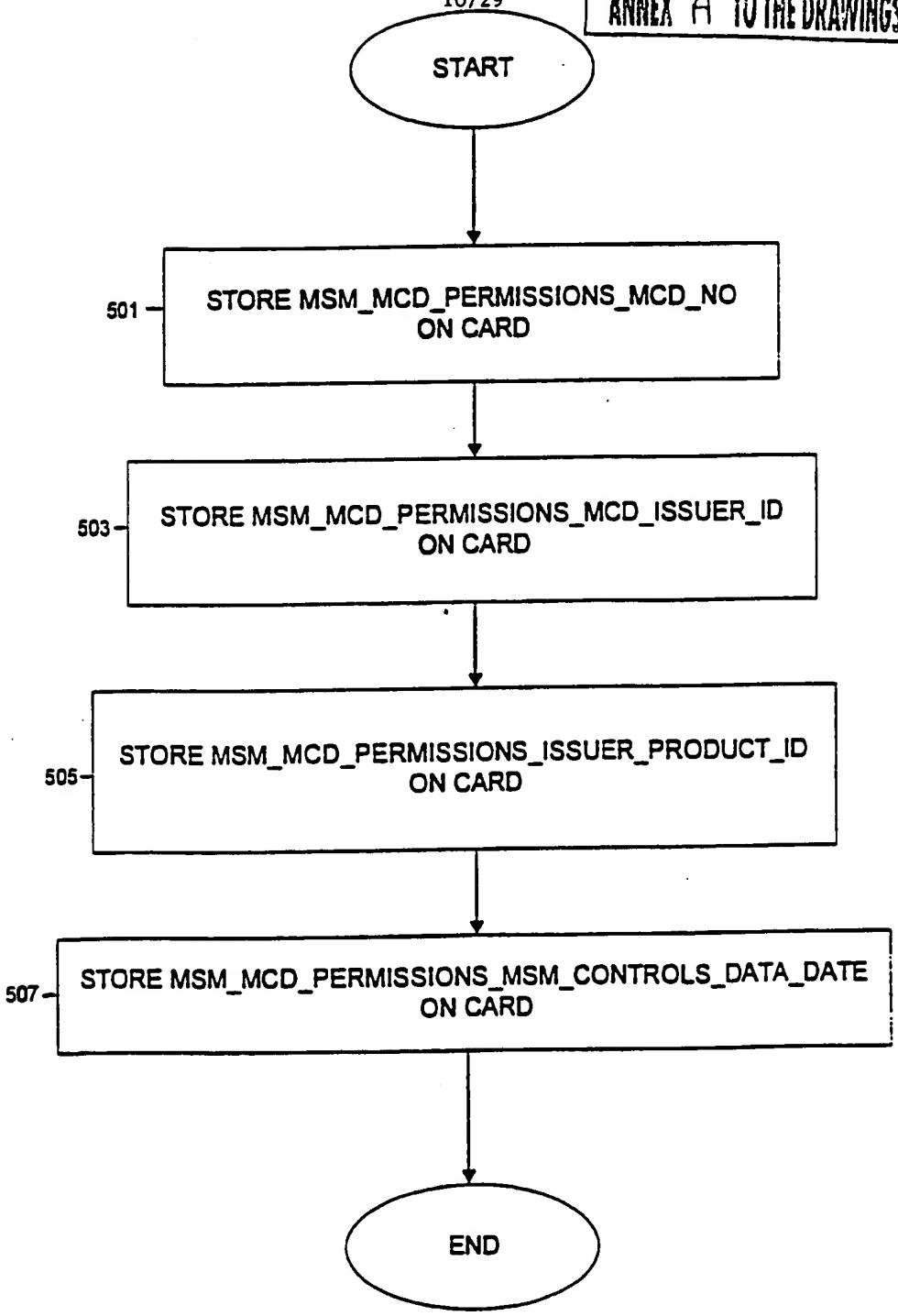
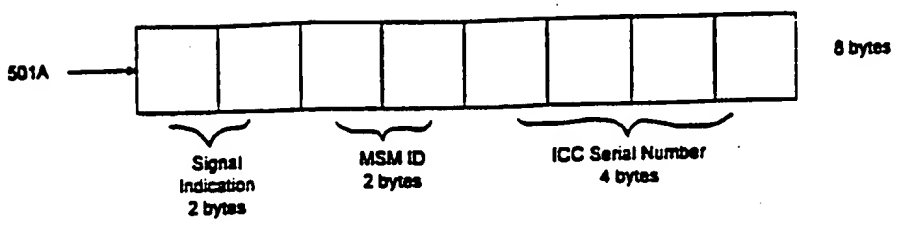
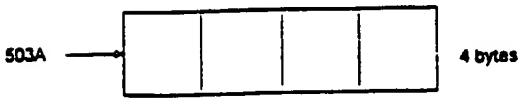
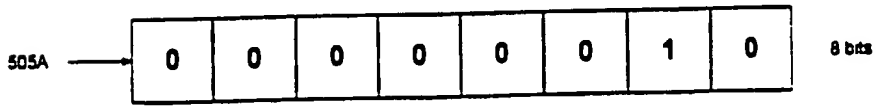


FIG. 5

ANNEX A TO THE DRAWINGS







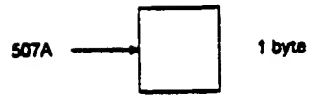


FIG. 5A

ANNEX A TO THE DRAWINGS

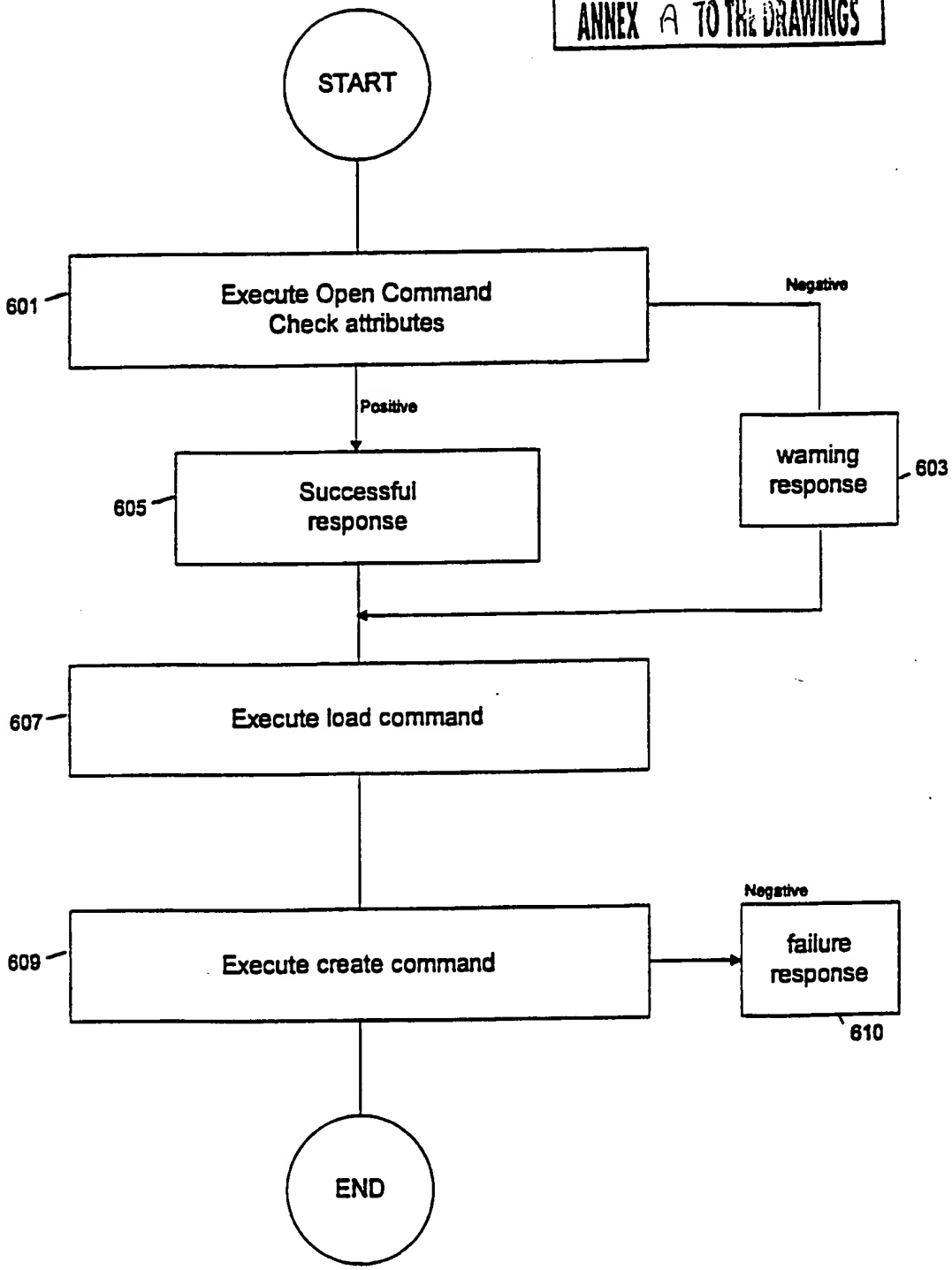


FIG. 6

ANNEX A TO THE DRAWINGS

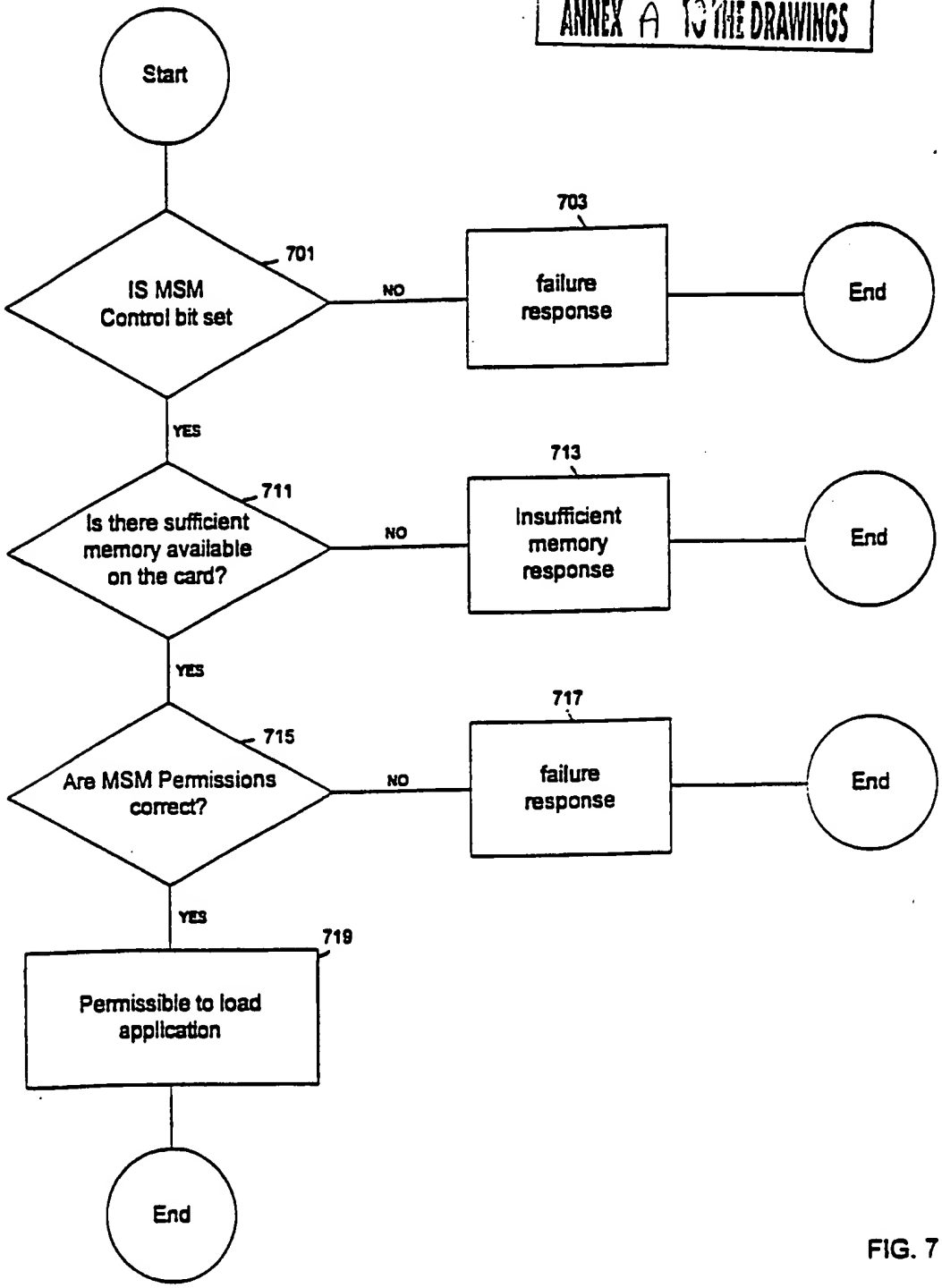


FIG. 7

ANNEX A TO THE DRAWINGS

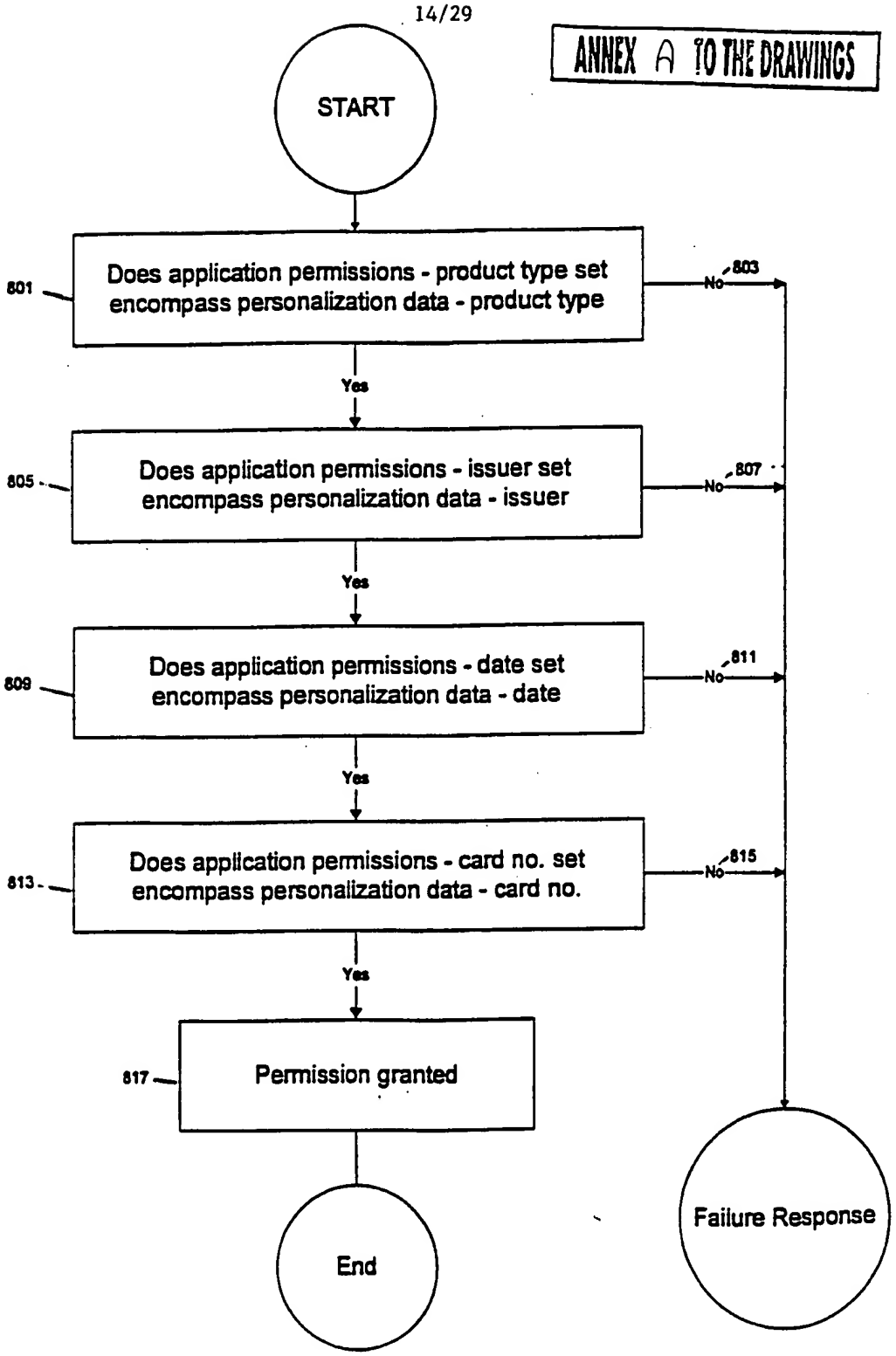


FIG. 8

ANNEX A TO THE DRAWINGS

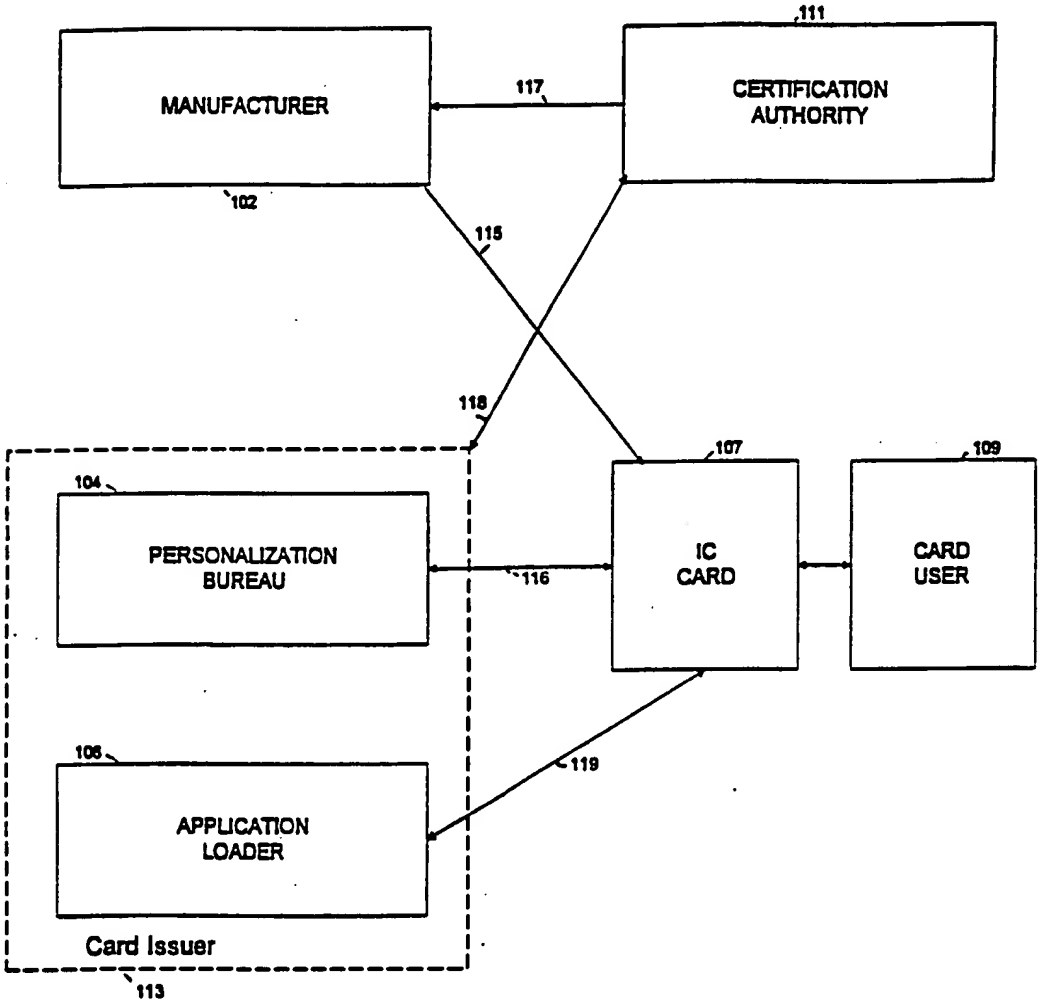


FIG. 9

ANNEX A TO THE DRAWINGS

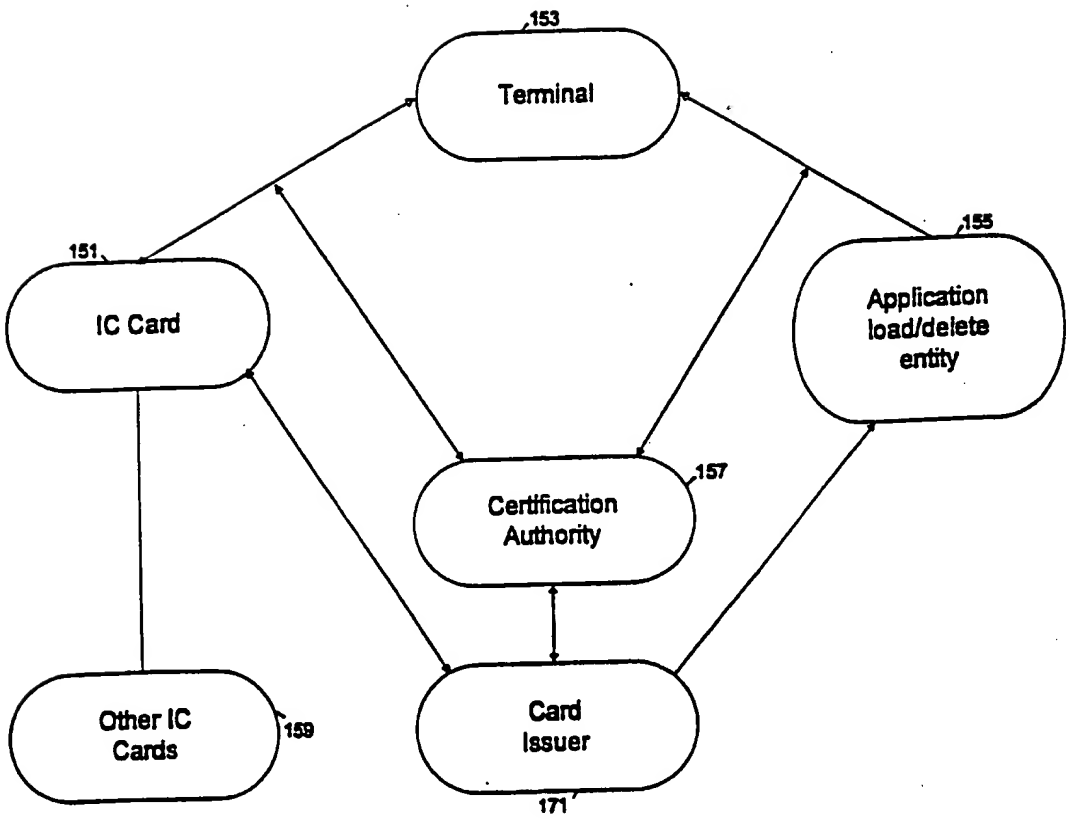


FIG. 10

ANNEX 6 TO THE DRAWINGS

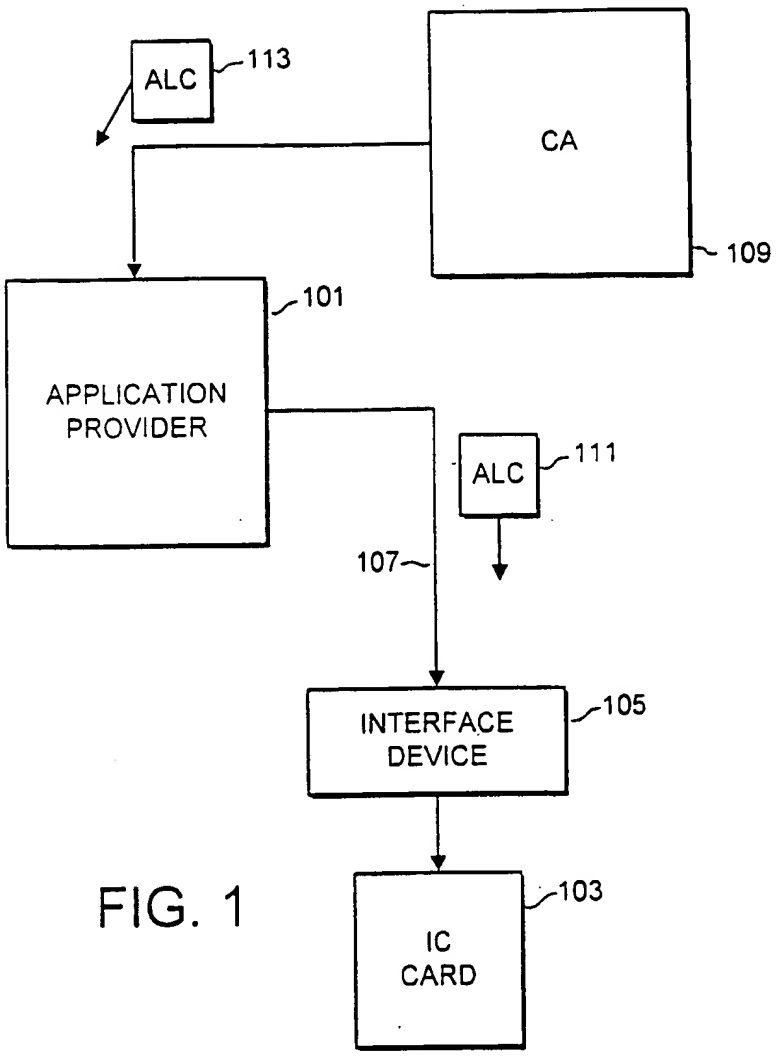


FIG. 1

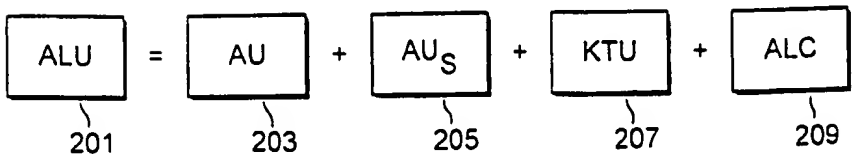


FIG. 2

ANNEX B TO THE DRAWINGS

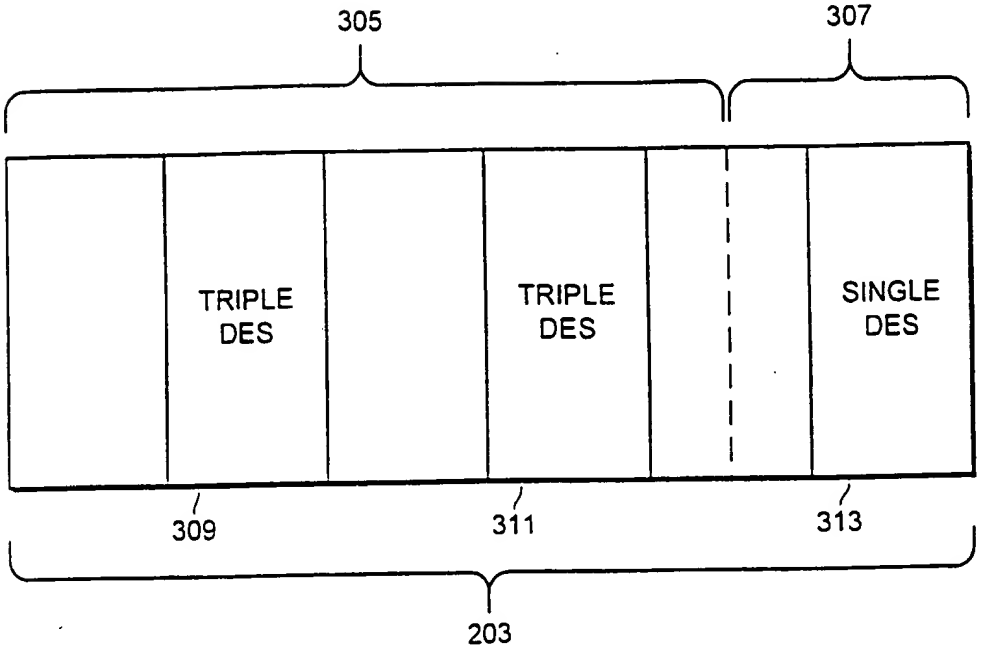


FIG. 3

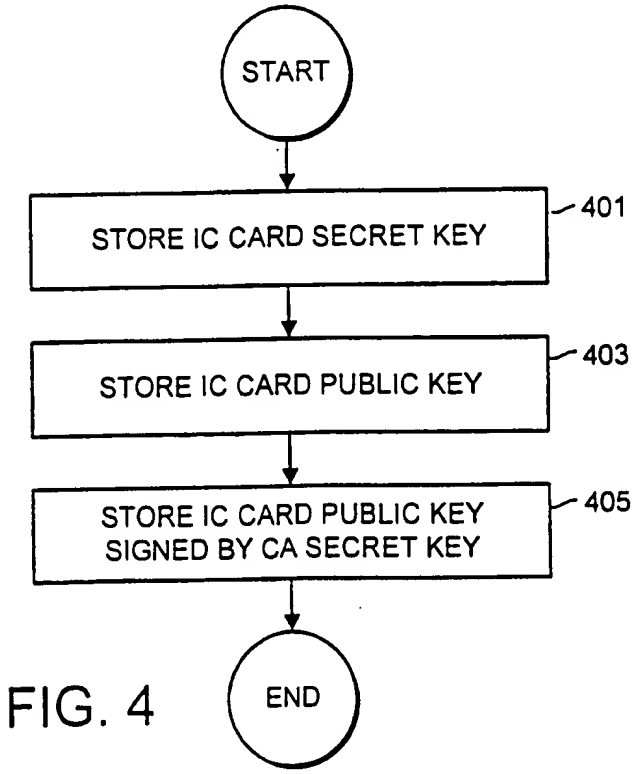


FIG. 4

ANNEX B TO THE DRAWINGS

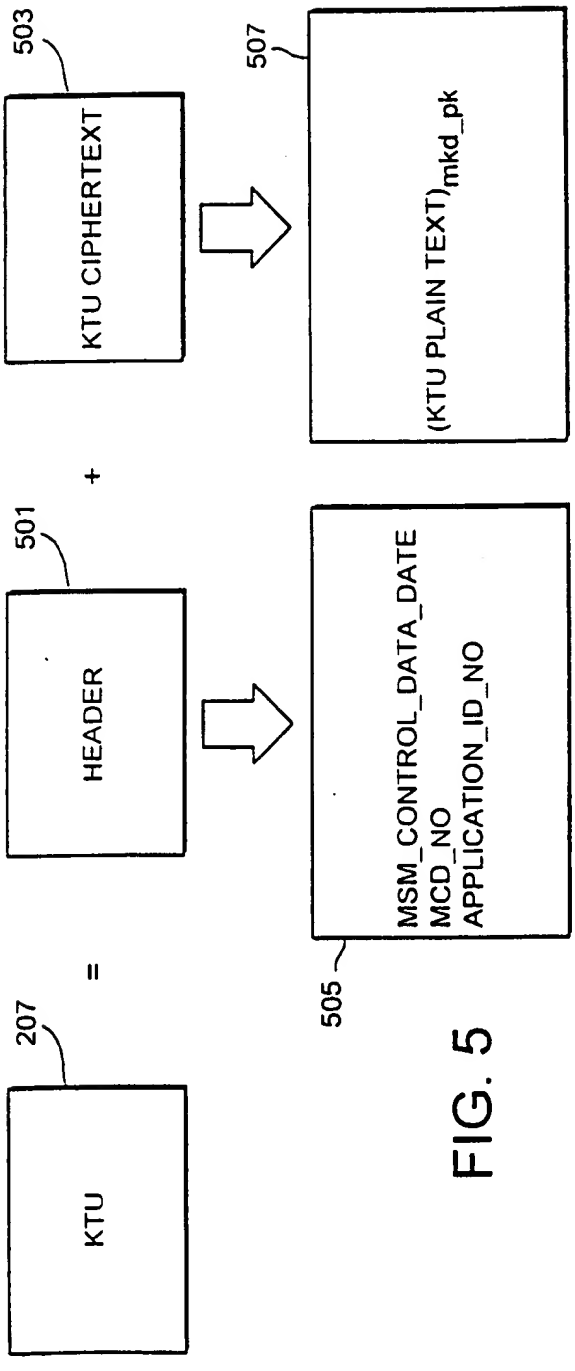


FIG. 5

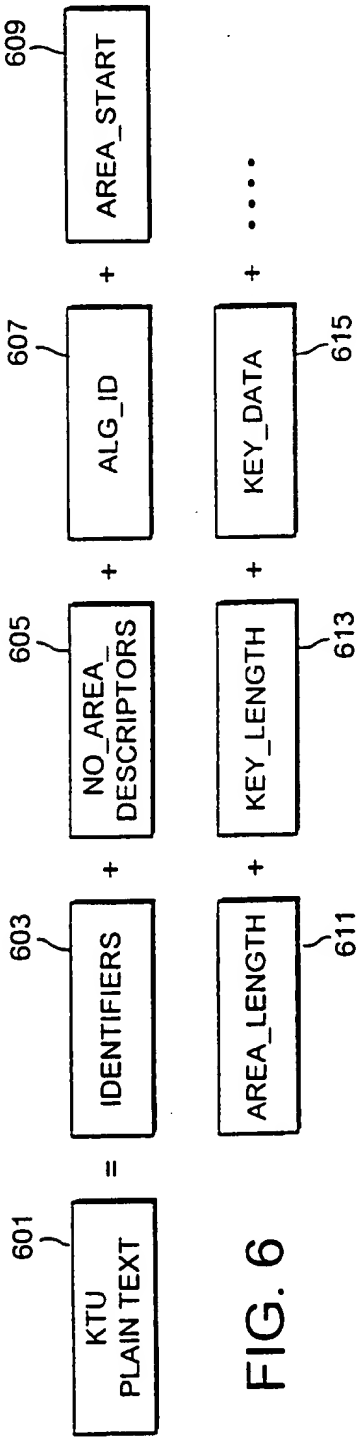


FIG. 6

ANNEX 6 TO THE DRAWINGS

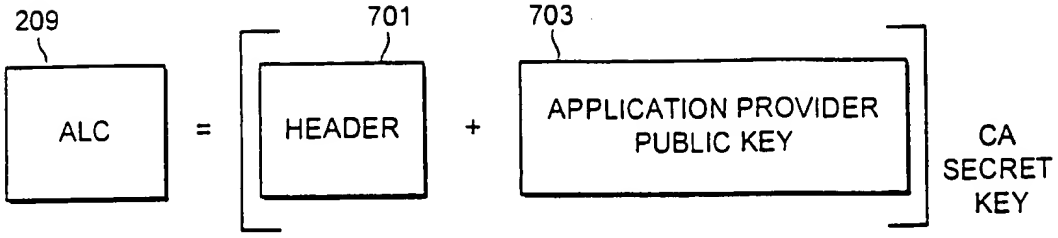


FIG. 7

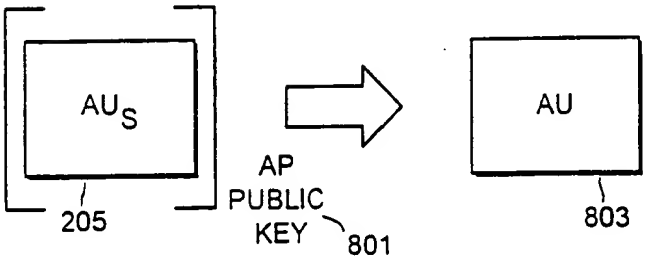


FIG. 8

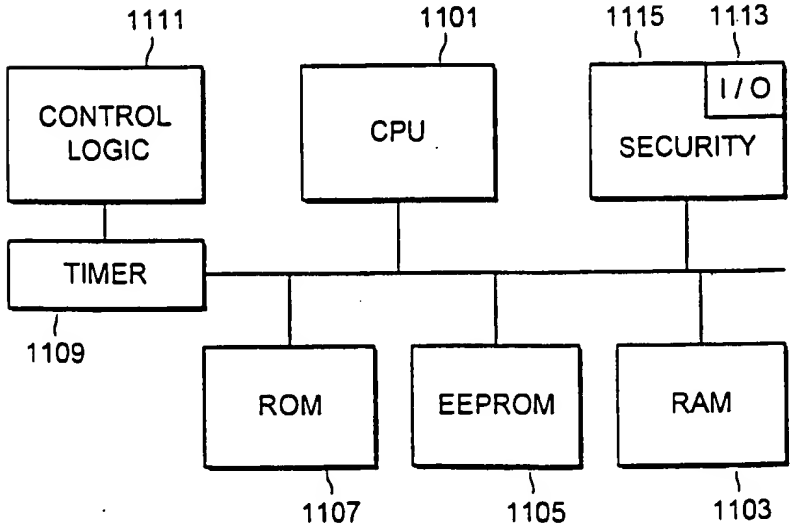


FIG. 11

ANNEX B TO THE DRAWINGS

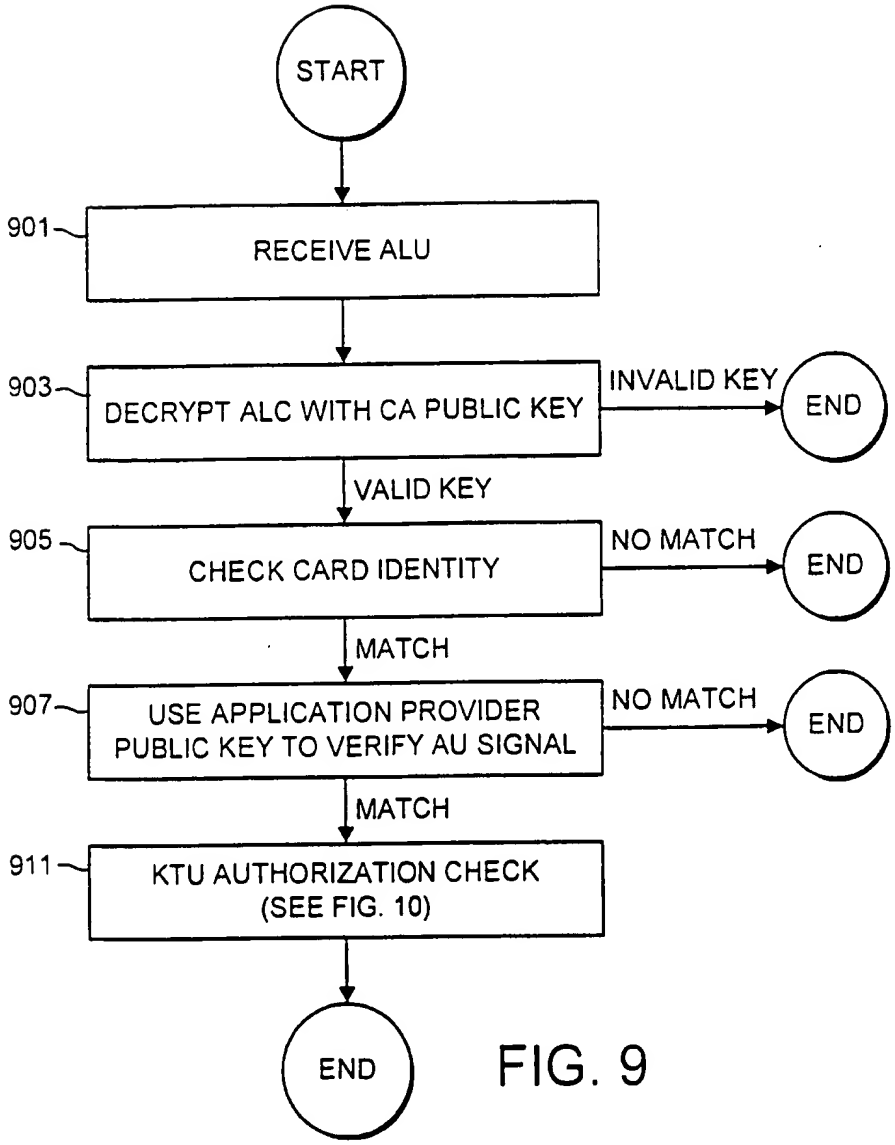


FIG. 9

ANNEX B TO THE DRAWINGS

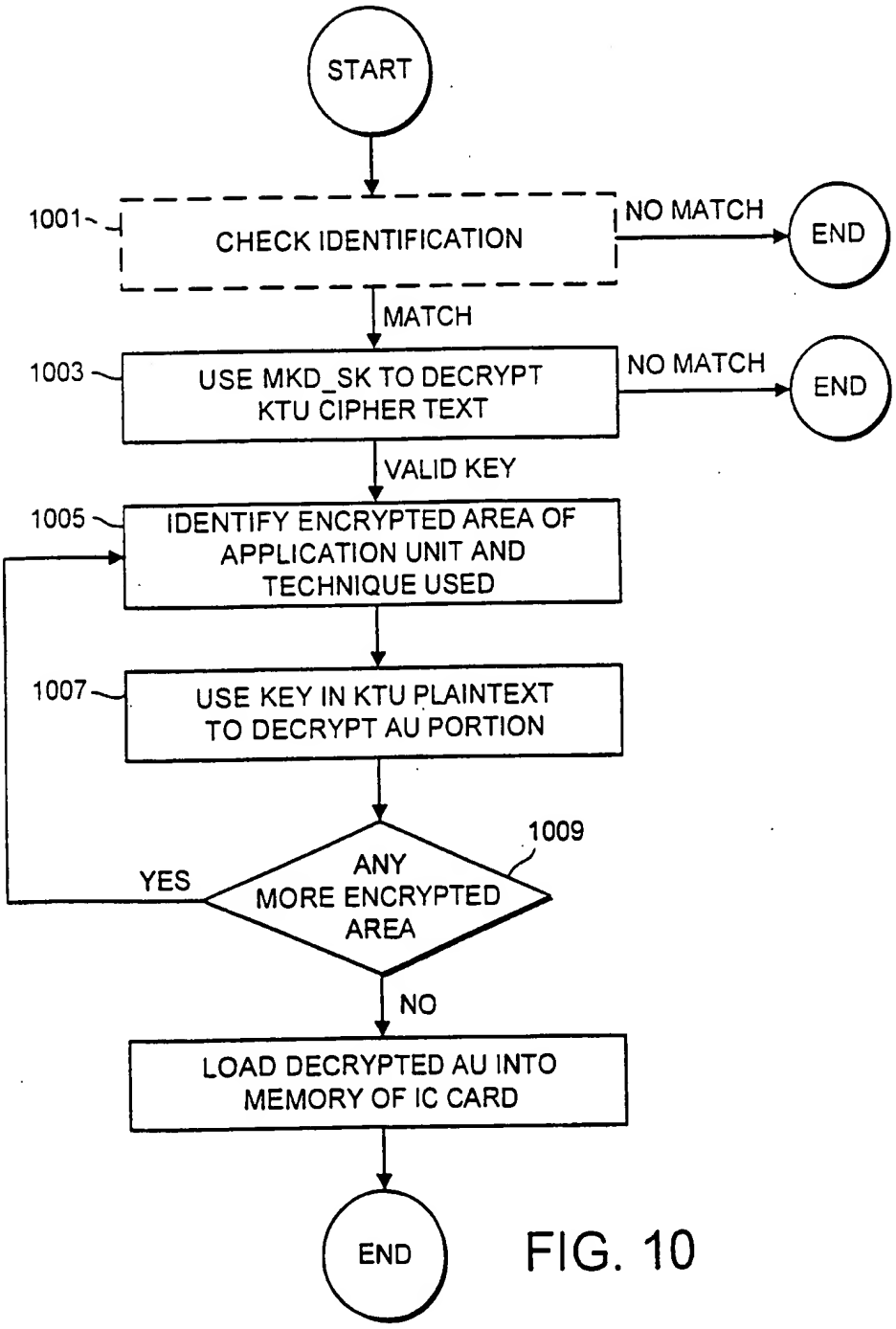


FIG. 10

ANNEX C TO THE DRAWINGS

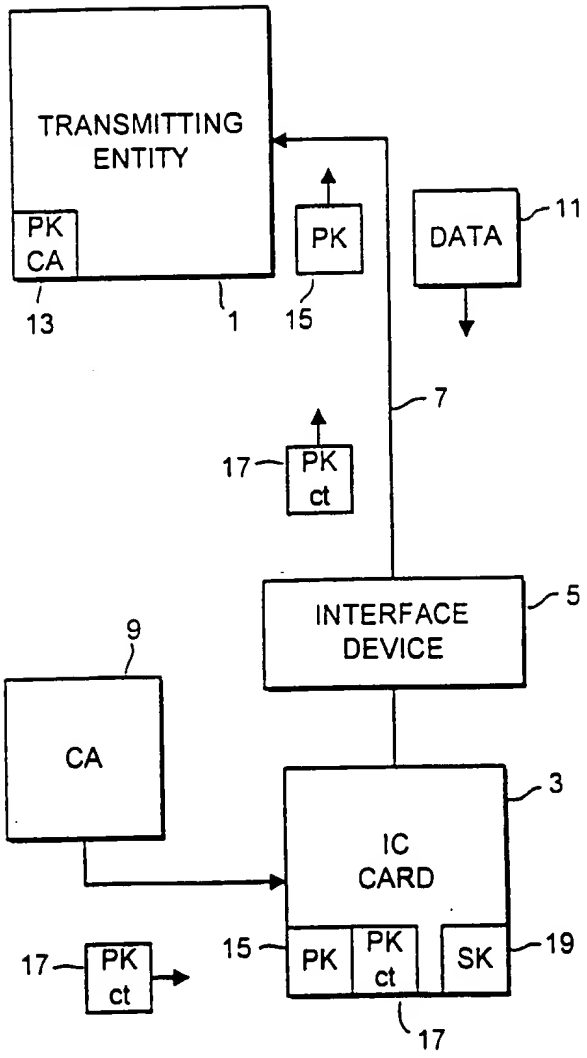


FIG. 1A

ANNEX C TO THE DRAWINGS

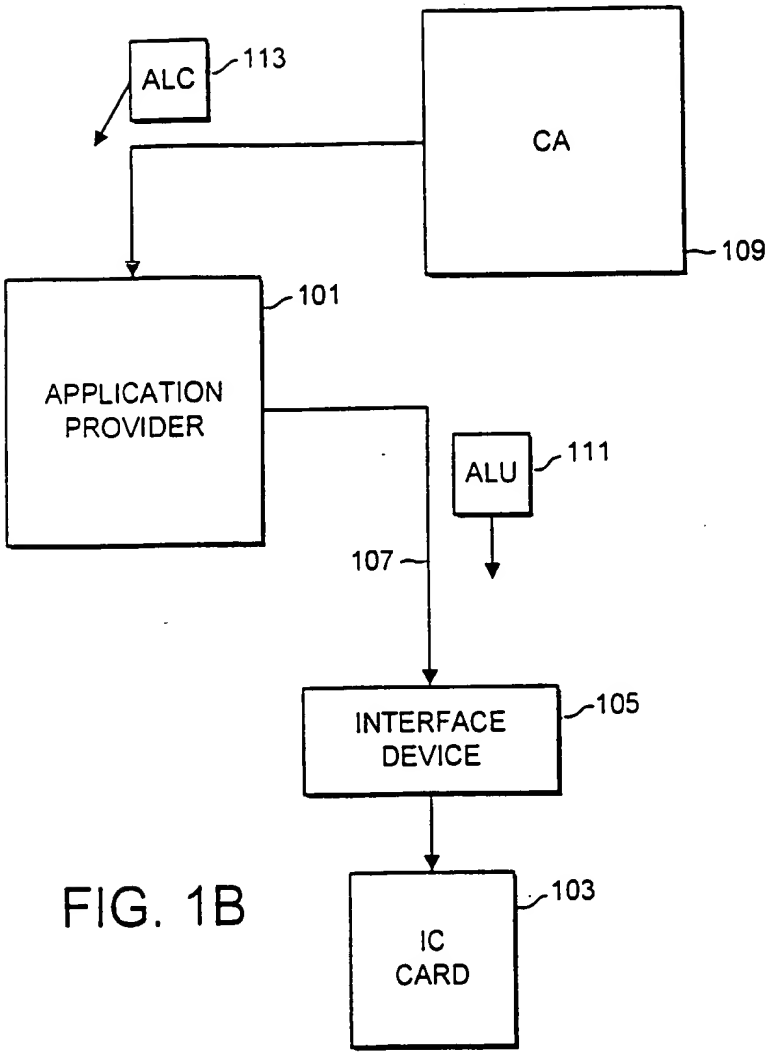


FIG. 1B

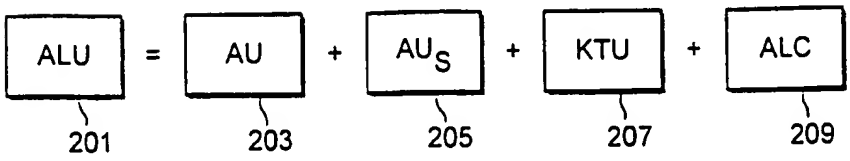


FIG. 2

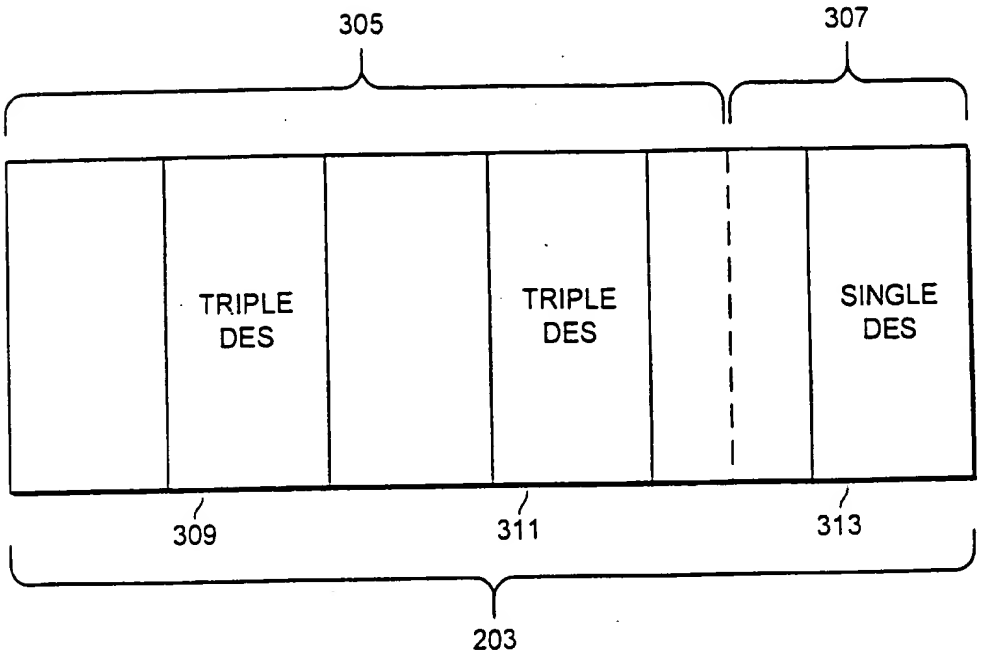


FIG. 3

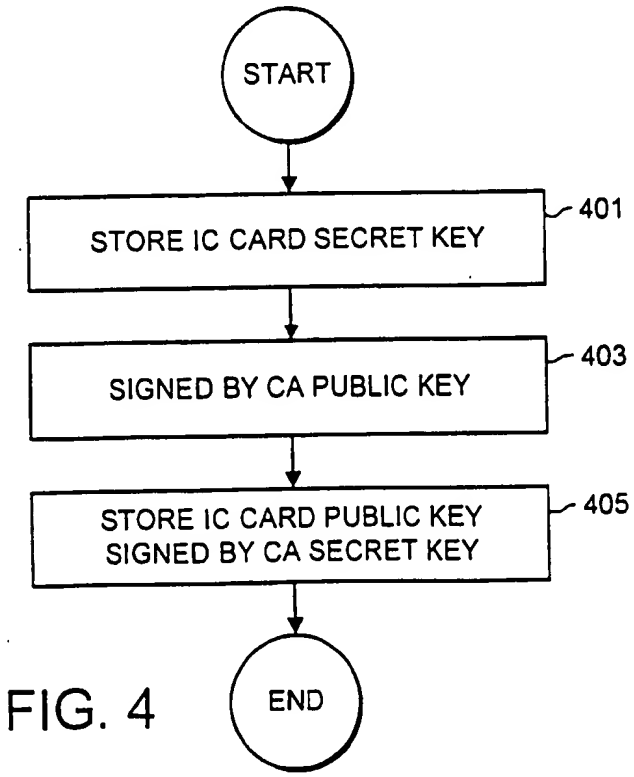


FIG. 4

ANNEX C TO THE DRAWINGS

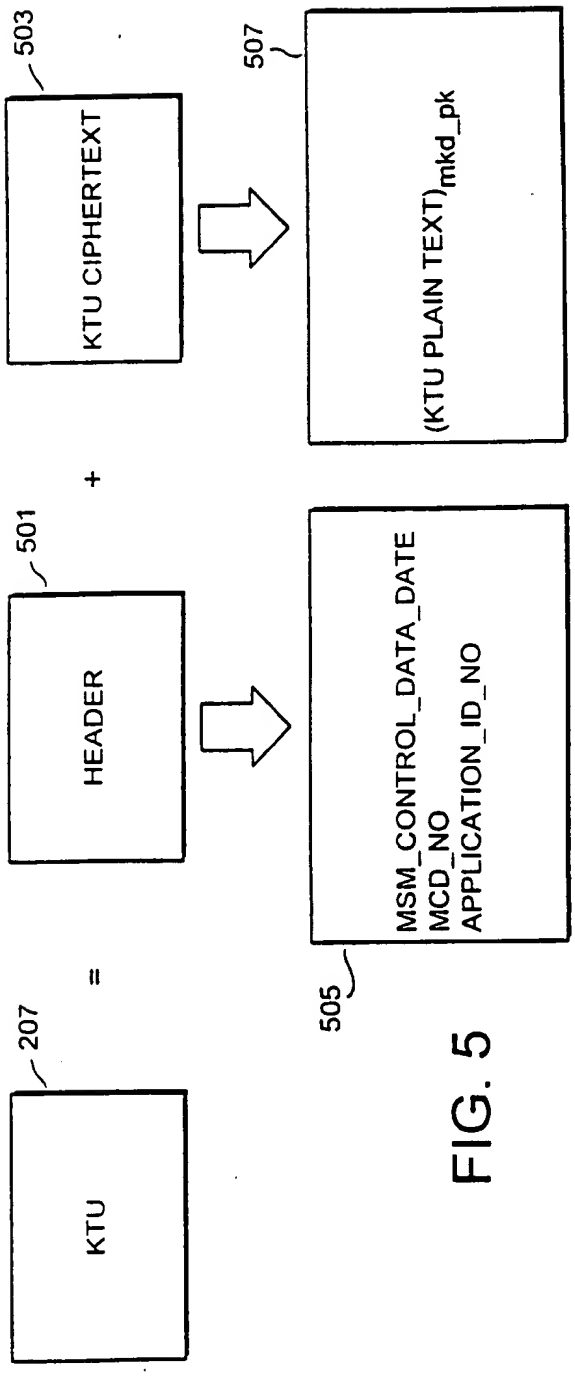


FIG. 5

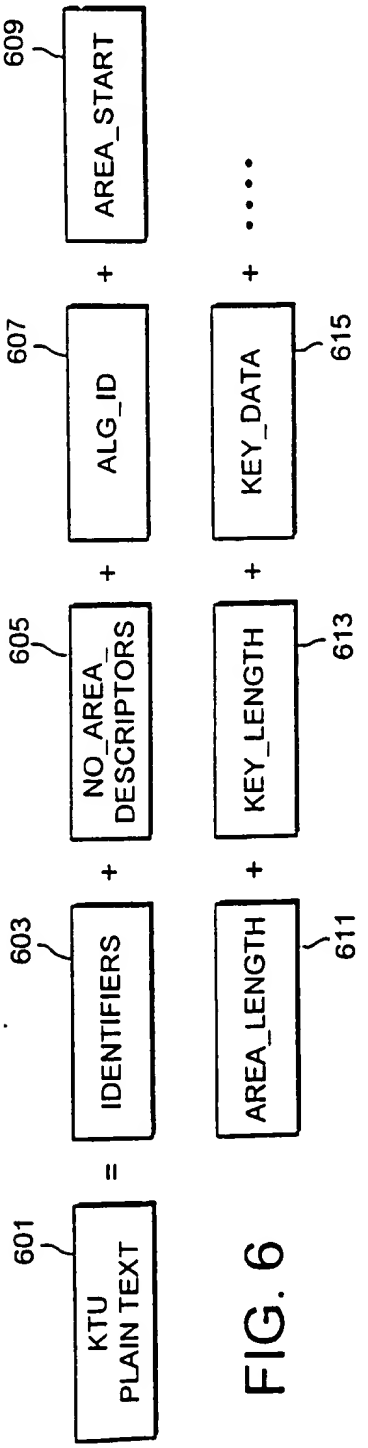


FIG. 6

ANNEX C TO THE DRAWINGS

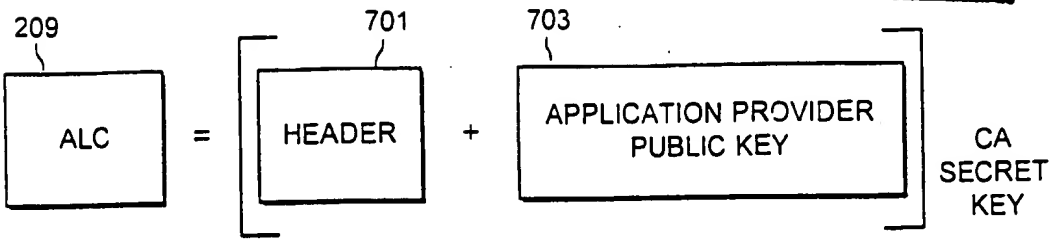


FIG. 7

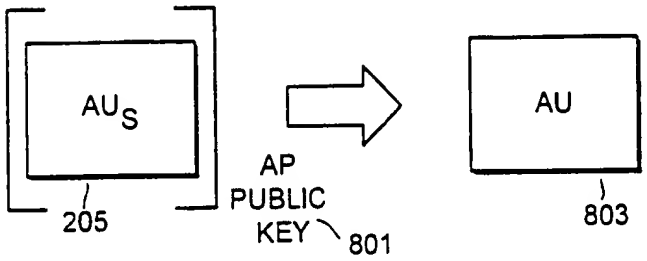


FIG. 8

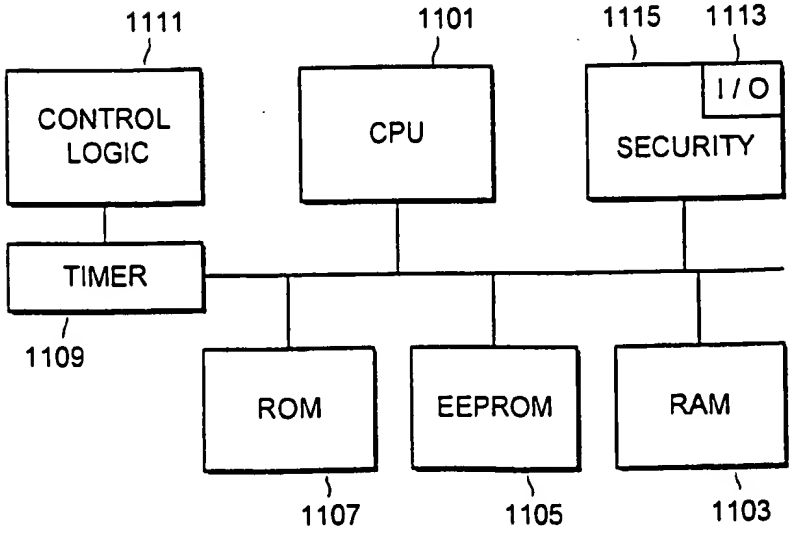


FIG. 11

ANNEX C TO THE DRAWINGS

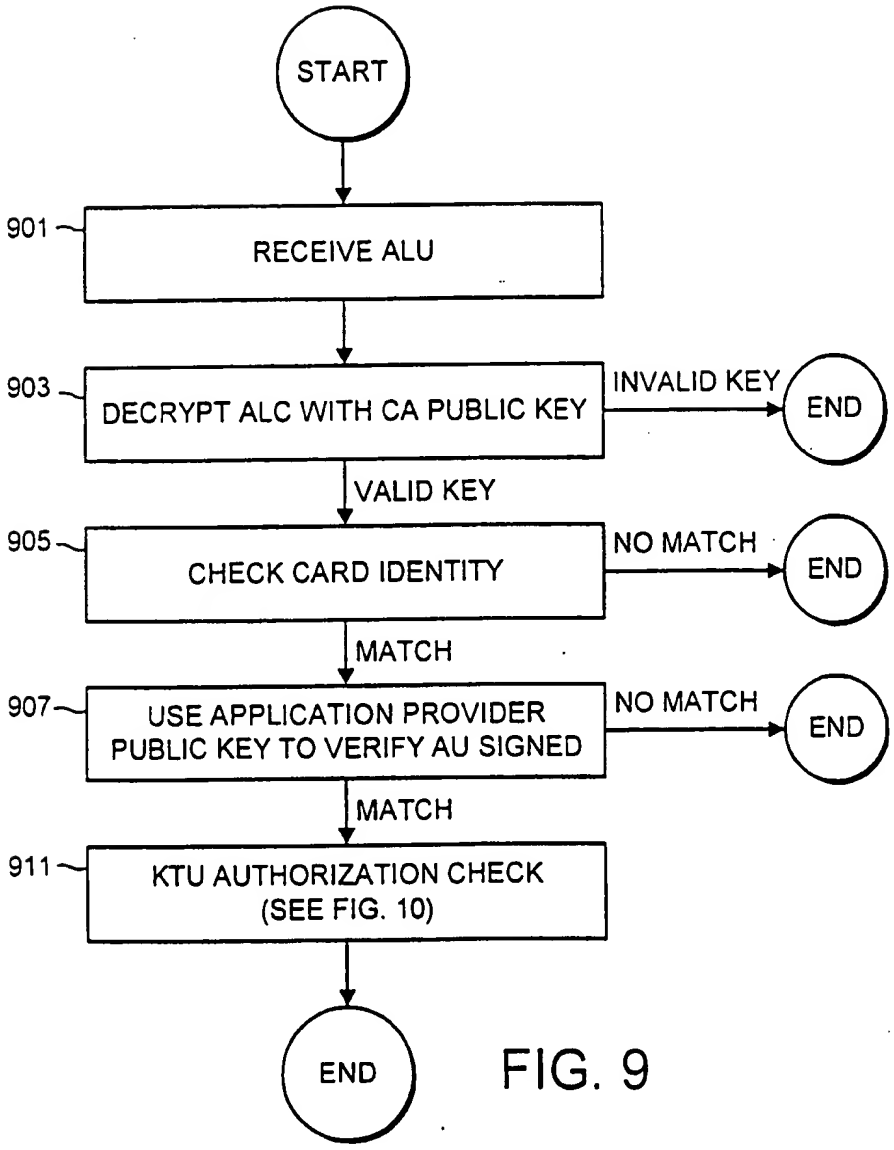


FIG. 9

ANNEX C TO THE DRAWINGS

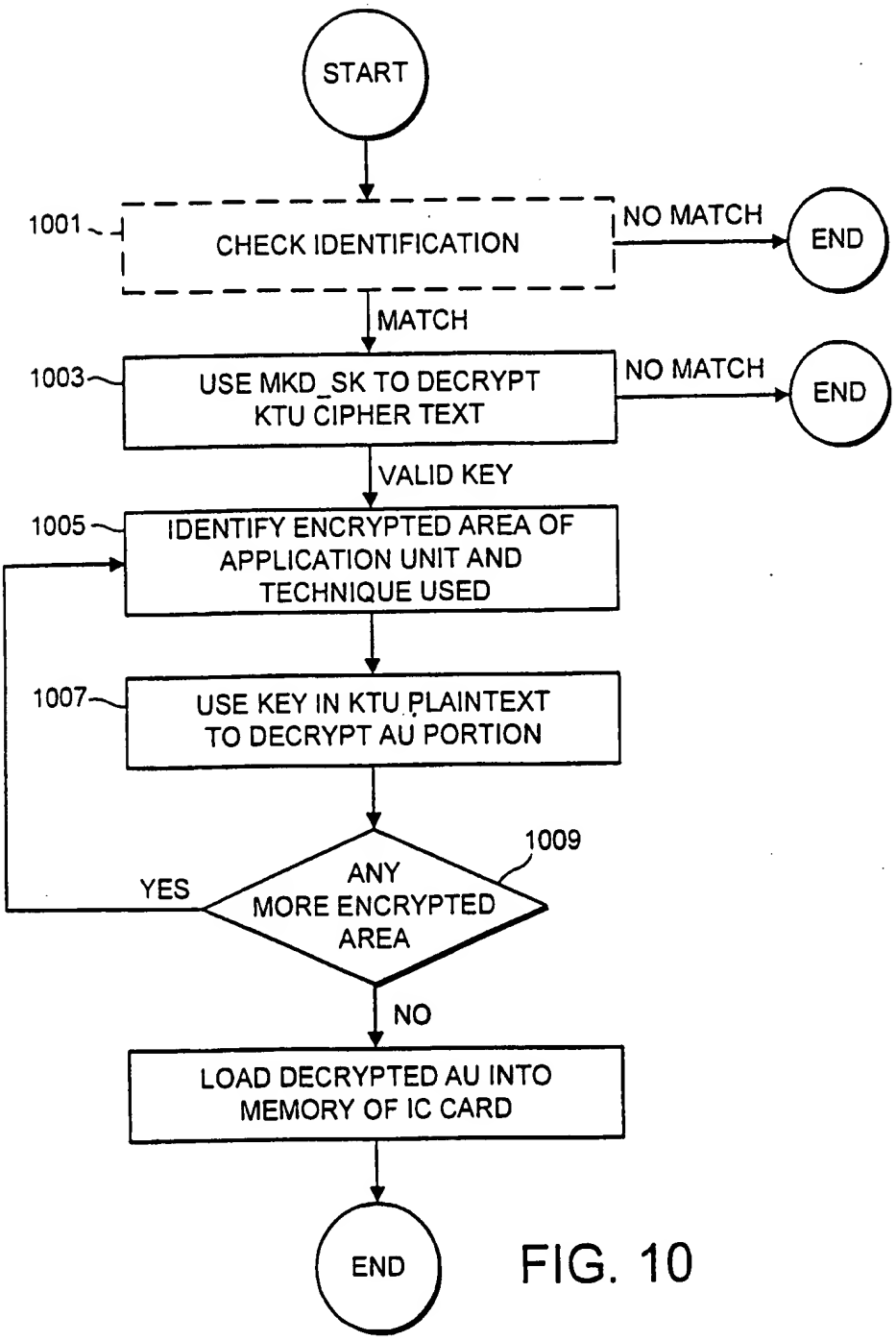


FIG. 10